
Directional Routing Techniques in VANET

PhD Thesis

Moath Muayad Al-Doori

This thesis is submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Software Technology Research Laboratory
De Montfort University
Leicester - United Kingdom

November 2011

Dedication

To my father

Professor Muayad Al-Doori

For his inspiration, Sacrifices, without his endless support I could not have
continued my studies.

To my mother

Nawar Al-Araji

For her endless love, support, encouragement, and ceaseless prayers.

To my wife

Zahra Al-Damluji

For her always being by my side, for her endless love.

Abstract

Vehicle Ad hoc Networks (VANET) emerged as a subset of the Mobile Ad hoc Network (MANET) application; it is considered to be a substantial approach to the ITS (Intelligent Transportation System). VANETs were introduced to support drivers and improve safety issues and driving comfort, as a step towards constructing a safer, cleaner and more intelligent environment. At the present time vehicles are equipped with a number of sensors and devices, including On Board Units (OBU); this enables vehicles to sense situations affecting other vehicles and manage communications, by exploiting infrastructures such as the Road Side Unit (RSU); creating a Vehicle to Infrastructure (V2I) pathway, or interacting directly with other vehicles creating a Vehicle to Vehicle (V2V) pathway. Owing to the lack of infrastructures and difficulties involved in providing comprehensive coverage for all roads because of the high expense associated with installation, the investigation in this research concentrates on the V2V communication type rather than the V2I communication type.

Many challenges have emerged in VANET, encouraging researchers to investigate their research in an attempt to meet these challenges. Routing protocol issues are considered to be a critical dilemma that needs to be tackled in VANET, particularly in a sparse environment, by designing an efficient routing mechanism that impacts on enhancing network performance in terms of disseminating messages to a desired

destination, balancing the generated packet (overhead) on the network and increasing the ratio of packet delivery with a reduced time delay. VANET has some unique characteristics compared to MANET; specifically it includes high mobility and constrained patterns restricted by roads, which lead to generation of a disconnected area occurring continuously between vehicles creating a Delay Tolerant Network (DTN). This is in opposition to applying the multi-hope technique properly to deliver the packet to its desired destination.

The aim in this thesis comprises two main contributions. First developing novel routing protocols for a sparse environment in VANET with the context of utilising the mobility feature, with the aid of the equipped devices, such as Global Position System (GPS) and Navigation System (NS). This approach exploits the knowledge of Second Heading Direction (SHD), which represents the knowledge of the next road direction the vehicle is intending to take, in order to increase the packet delivery ratio, and to increase the route stability by decreasing instances of route breakage. This approach comprises two approaches; the first approach was designed for a highway scenario, by selecting the next hop node based on a filtration process, to forward the packet to the desired destination, while the second approach was developed for the intersection and roundabout scenario, in order to deliver the packet to the destination (unknown location).

The formalising and specification of the VSHDRP has been performed using the CCA (Calculus of Context-aware Ambient), in order to evaluate the protocols behaviours, the protocol has been validated using the ccaPL. In addition the performance of the VSHDRP has been evaluated using the NS-2 simulator; comparing it with Greedy Perimeter Stateless Routing (GPSR) protocol, to reveal the strengths and weaknesses of the protocol.

Second, developing a novel approach to broadcasting the HELLO beacon message adaptively in VANET based on the node's circumstances (direction and speed), in order to minimise the broadcasting of unnecessary HELLO beacon messages. A novel architecture has been built based on the adaptive HELLO beacon message, which clarifies how the OBU components are interacting with the connected sensors, in order to portray any changes in the vehicle's circumstances, so as to take the right decision to determine appropriate action. This architecture has been built based on the concept of a context aware system, which divides the architecture into three main phases; sensing processing and acting.

Declaration

I declare that the work described in this thesis is original work undertaken by me for the degree of Doctor of Philosophy, at the software Technology Research Laboratory (STRL), at De Montfort University, United Kingdom.

No part of the material described in this thesis has been submitted for any award of any other degree or qualification in this or any other university or college of advanced education.

This thesis is written by me and produced using L^AT_EX.

Moath Al-Doori

Publications

1. Moath M. Al-Doori, Ali H. Al-Bayatti and Hussein Zedan. Context Aware System Architecture For Sending Adaptive HELLO Message in VANET. *In Proceeding of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems, CASEMANS'10* - Copenhagen, Denmark, pages 65-68, 2010.
2. Moath M. Al-Doori, François Siewe and Ali H. Al-Bayatti. VSHDRP:Vehicle Second Heading Direction Routing Protocol in VANET. *In Proceeding of the 4th IEEE International Conference on Computer Science and Information Technology, ICCSIT 2011* - Chengdu, China, pages 381-386, 2011.
3. Moath M. Al-Doori, François Siewe and Ali H. Al-Bayatti. *Routing Management for DTN Network in VANET*. International Journal of Machine Learning and Computing (IJMLC), 2011.(accepted).

Acknowledgments

First and foremost, I am thankful to almighty Allah (God), for giving me the ability to complete this research, without whom nothing is possible.

My sincerest gratitude and deepest appreciation goes to Prof. Hussein Zedan, the head of the STRL for his guidance throughout my PhD study. I also greatly appreciate his care and concern throughout my PhD study.

I would like to thank Dr. Ali Al-Bayati for his thoughtful and valuable guidance, insightful suggestions in the research work and constant encouragement.

I would like to thank Dr. Francois Siewe for his expert guidance and technical helps and suggestions. I am grateful for his careful reading and constructive comments on our joint papers.

I would like to thank all researchers, colleagues and staff of the Software Technology Research Laboratory (STRL), Faculty of Technology, De Montfort University for the friendly and convenient working atmosphere I could experience there.

On family side, I have not enough words to thank my ever first teachers in this life, my parents, who have been inspiring me and pushing me forward all the way.

Their prayers and blessings were no doubt the true reason behind any success I have realized in my life.

I would also like to thank my beloved wife, Zahra Al-Damluji who has always been by my side giving me her love, encouragement, and support through my tough times, I thank her for the limitless patience she used always to show and for the times she backed me in the family being the mother and father at the same time.

My thanks and loves to my beautiful daughter Hala and my beloved Son Yusif, who have always pushing me forward through their love.

My special gratitude is due to my dearest brother Mohammed and to my lovely sisters Tara and Nadia for their loving support, concern and encouragement through all these years.

I would like to thanks my father in law Dr. Nawfal Al-Damluji and my mother in law Mrs. Reem Al-Jalili for their endless support and concern.

Contents

Dedication	I
Abstract	II
Declaration	V
Publications	VI
Acknowledgments	VII
TABLE OF CONTENTS	XV
LIST OF FIGURES	XVIII
LIST OF TABLES	XIX
List of Abbreviations	XX
1 Introduction	1
1.1 Research Motivations	2
1.2 Problem Description and Routing Solution	3
1.3 Aims of the Research	5
1.4 Contributions	6
1.5 Thesis Outlines	7

2	Overview of Routing Protocols in VANET	10
2.1	Introduction and Motivation	11
2.2	Why MANET routing protocols are not suitable for VANET scenarios	12
2.3	Classification of Routing Protocols in VANET	14
2.3.1	Topology-Based Routing Protocols	16
2.3.1.1	Proactive Routing Protocols (periodic)	16
2.3.1.2	Reactive Routing Protocols (On-Demand)	17
2.3.1.3	Hybrid	18
2.3.2	Geographic-Based Routing Protocols (Position-Based)	19
2.3.2.1	Non-DTN Geographic Routing Protocols	20
2.3.2.2	DTN-Geographic Routing Protocols	27
2.4	Why Focusing on DTN Networks	40
2.5	Summary	41
3	Preliminaries	42
3.1	Research Introduction and Motivation	43
3.2	Research Methodology	45
3.3	The Key Concept of the Research	46
3.3.1	Adaptive HELLO Message in VANET	46
3.3.2	Vehicle Second Heading Direction Routing Protocol in VANET (VSHDRP)	47
3.3.2.1	The principles of VSHDRP1	48
3.3.2.2	The principles of VSHDRP2	49
3.3.3	VSHDRP Model Definition	49
3.3.3.1	System Model Assumptions	50
3.3.3.2	Network Models	51

3.3.3.3	Mobility Features	55
3.3.3.4	Traffic Model	55
3.4	Overview of context aware system	56
3.5	The Stages of Context-aware System	57
3.5.1	Sensing Stage	58
3.5.2	Thinking Stage	59
3.5.3	Acting Stage	60
3.6	VSHDRP Routing Protocol Evaluation	60
3.6.1	CCA Based Evaluation	60
3.6.1.1	The Calculus of Context-aware Ambients(CCA)	60
3.6.2	NS-2 Based Evaluation	65
3.6.2.1	Network Simulator-2 (NS-2)	66
3.7	Simulation Environment of the Routing Protocol Mechanism	68
3.8	Summary	70
4	On Board Unit Architecture Based on Context-aware System	72
4.1	Introduction and Motivation	73
4.2	Overview of the adaptive HELLO Message	74
4.2.1	Message Broadcast Conditions	75
4.2.2	Adaptive HELLO Beacon Message Scenario	77
4.2.3	Adaptive HELLO Message Mechanism	77
4.3	OBU Architecture	79
4.3.1	Collective Knowledge Phase	81
4.3.2	Processing Phase	83
4.3.3	Reaction Phase	85
4.4	Summary	85

5	Vehicle Second Heading Direction Routing Protocol (VSHDRP1)	86
5.1	Introduction and motivation	87
5.2	VSHDRP1 Routing Protocol	90
5.3	Mobility knowledge Acquisitions	93
5.4	VSHDRP1 Mechanism	94
5.4.1	Packet sending process	94
5.4.2	Filtration process for selecting the next hop node	95
5.4.2.1	Filtration Process Stages	96
5.4.2.2	Filtration Process Algorithm	97
5.4.3	Packet Delivery Confirmation	97
5.5	Example of Packet Propagation	98
5.6	Formal Specification of VSHDRP1	100
5.6.1	System Model	102
5.6.2	Context Expression	103
5.6.3	Source Node	104
5.6.4	Intermediate Node	106
5.6.5	Destination Node	107
5.7	Behavioural Validation and Analysis based on CCA	107
5.7.1	Scenario 1	108
5.7.2	Scenario 2	109
5.7.3	Scenario 3	110
5.8	Straight Highway Scenario	112
5.9	Simulation Methodology and Model	114
5.10	Simulation Environment	115
5.11	Parameter Values	116
5.12	The Result and Analysis of VSHDRP1 Routing Protocol simulation .	117
5.13	Simulation Metrics	117

5.14	Result and Analysis	118
5.14.1	The Efficiency of Data Packet Delivery Ratio	118
5.14.1.1	Network Size	119
5.14.1.2	Data Packet Size	120
5.14.1.3	HELLO beacon Message Period Intervals	121
5.14.2	Generated Data Traffic (Overhead)	122
5.14.2.1	Network Size	123
5.14.2.2	Data Packet Size	124
5.14.2.3	HELLO beacon Message Period Intervals	125
5.14.3	The Time Delay of Data Delivery	125
5.14.3.1	Network Size	126
5.14.3.2	Data Packet Size	126
5.14.3.3	HELLO beacon Message Period Intervals	127
5.15	Summary	129
6	Vehicle Second Heading Direction Routing Protocol (VSHDRP2)	131
6.1	Introduction and Motivation	132
6.2	The proposed Routing Protocol VSHDRP2	133
6.2.1	Mobility and Nodes Classification	133
6.2.2	Packet Sending Process	135
6.2.3	Packet Delivery Confirmation	138
6.3	Example of packet propagation process in VSHDRP2	139
6.4	Formal Specification of VSHDRP2	142
6.4.1	System Model	143
6.4.2	Context Expression	144
6.4.3	Source Node	145
6.4.4	Intermediate Node	147

6.4.5	Destination Node	148
6.4.6	Analysis of the Protocol	148
6.4.6.1	Scenario 1	149
6.4.6.2	Scenario 2	151
6.4.6.3	Scenario 3	152
6.5	Roundabout/Intersection Scenario	152
6.6	NS-2 Simulation	154
6.7	Summary	154
7	Conclusion and Future Work	156
7.1	Conclusion	157
7.2	Achieved Aims	159
7.3	Future work	160
A	Overview in VANET	175
A.1	History and Background of MANET	175
A.2	MANET Characteristics	176
A.3	Types of applications in MANET	179
A.4	VANET Overview	180
A.5	Vehicle Communication Categories	180
A.5.1	In-Vehicle Communications (InVC)	181
A.5.2	Inter-Vehicle Communication (IVC)	182
A.5.3	Vehicle-to-Infrastructure Communication	183
A.6	VANET Characteristics	185
A.7	Applications in VANET	188
A.7.1	Entertainment/ Information Applications	189
A.7.2	Safety Applications	189
A.8	Wireless Access Technology in VANET	191

A.9 Challenges and Requirements in VANET	196
B Formal specification in the Calculus of Context-aware Ambients(CCA) for VSHDRP1	199
C Formal specification in the Calculus of Context-aware Ambients(CCA) for VSHDRP2	219

List of Figures

2.1	Classification of various routing protocol in VANET	15
3.1	The collision of packets in simultaneous transmission	54
3.2	Stages of context aware system	58
3.3	Routing mechanism evaluation	61
3.4	Simulator usage from MobiHoc survey [1]	67
3.5	NS-2 schematics structure [2]	68
4.1	Broadcasting when the vehicle leaves the road	75
4.2	Broadcasting when the vehicle merges in the road	76
4.3	Sending HELLO message adaptively	79
4.4	OBU architecture for adaptive HELLO message based on context-aware system	80
4.5	The three phases architecture	81
4.6	Collective knowledge phase	82
4.7	Active safety applications	83
5.1	VSHDRP1 algorithm	95
5.2	Filtration process diagram for selecting the next-hop node	96
5.3	Filtration process algorithm for selecting next-hop node	98
5.4	Next-hop node selection according to the neighbour's positions	99

5.5	Next-hop node selection according to neighbour's current direction . .	100
5.6	Next-hop node selection according to neighbour's opposite direction .	100
5.7	CCA Scenario1	108
5.8	CCA Scenario 2	109
5.9	CCA Scenario3	111
5.10	Highway Scenario	114
5.11	The efficiency of data packet delivery against the nodes number (network size)	119
5.12	The efficiency of data packet delivery against the data packet size . .	121
5.13	The efficiency of data packet delivery against the beacon message intervals	122
5.14	Overhead against network size	123
5.15	The data packet size against the number of packets	124
5.16	Overhead against the beacon message intervals	125
5.17	Time delay against network size	127
5.18	Time delay against data packet size	128
5.19	Time delay against interval of HELLO beacon message	128
6.1	The classification of neighbour nodes according to the SHD angles . .	134
6.2	VSHDRP2 algorithm	136
6.3	Filtration process for selecting the next-hop node	138
6.4	The node distribution based on the SHD	139
6.5	Next hop-node selection based on position	141
6.6	Next hop-node selection based on SHD	142
6.7	Roundabout Scenario	153
A.1	Ad hoc network of four nodes, node A communicates with node D . .	177
A.2	Types of communication in VANET [3]	181

A.3	Inter-vehicle Ad hoc (V2V) communication type [4]	183
A.4	V2I communication type	184
A.5	Avoiding crashes with the aid of V2V and V2I [5]	188
A.6	Scenario of combined types of communications [6]	194
A.7	DSRC channels spectrum [7]	195

List of Tables

2.1	Requirements of routing protocols in DTN VANET	41
3.1	the standard HELLO beacon message format in VANET	51
3.2	Syntax of CCA [8]	62
3.3	Syntax of contexts	64
3.4	NS-2 simulation Setup	70
5.1	HELLO beacon message format of VSHDRP1	93
5.2	Information of nodes in the network	101
5.3	Route path one	101
5.4	Route path two	101
5.5	Output of CCA scenario 1	108
5.6	Output of CCA scenario 2	110
5.7	Output of CCA scenario 3	111
5.8	NS-2 simulation Setup	117
6.1	HELLO beacon message format for VSHDRP2	135
6.2	Output of scenario 1	150
6.3	Output of scenario 2	151
6.4	Output of scenario 3	152

List of Abbreviation

ABR	Associativity-Based Routing
AGF	Advance Greedy Forwarding
AODV	Ad hoc On-demand Distance Vector
ARPA	Advanced Research Project Agency
AU	Application Units
BS	Base Stations
CAR	Connectivity-Aware Routing
CBF	Contention-Based Forwarding
CBR	The constant Bit Rate
CCA	Calculus of Context-aware Ambient
CCH	control channel
CEDAR	Core Extraction Distributed Ad hoc Routing
CTD	Conrmation Time Duration
DARPA	Defence Advanced Research Project Agency
DCMU	Direction Control and Monitor Unit

DL	DownLink
DN	Destination Node
DoS	Denial of Service
DSDV	Destination Sequence Distance Vector
DSRC	Dedicated Short Range Communication
DSR	Dynamic Source Routing
DTN	Delay Tolerant Network
ertPS	extended real time Polling Service
ETC	Electronic Toll collection
FCC	Federal Communication Commission
FSR	Fisheye State Routing
GeOpps	Geographical Opportunistic Routing
GloMoSim	global Mobile System Information System Simulator Library
GPCR	Geographic Routing in City Scenarios
GPS	Global Position System
GSR	Global State Routing
GPSR	Greedy Perimeter Stateless Routing
GUI	Graphical User Interface
ICT	Information and Communication Technology
IfQ	Interface Queue
IMANET	Internet-Based Mobile Ad Hoc Networks
IMT	International Mobile Telecommunication

IN	Intermediate Node
InVC	In-Vehicle Communication
IR	infrared
ITS	Intelligent Transportation System
ITUR	International Telecommunication Union-radio communication
IVC	Inter-Vehicle Communication
LDU	Link Delay Update
LET	link expiration time
LL	Link Layer
LOS	line of sight
MANET	Mobile Ad hoc NEtwork
MIMO	Multi Input Multi Output
MOVE	Motion Vector Routing Algorithm
MPDD	Multi- Path Data Dissemination
Nb	neighbour node
NS	Navigation System
NS-2	Network Simulator version-2
nrtPS	non real time Polling Service
OBU	On Board Units
OFDMA	Orthogonal Frequency Division Multiple Access
OLSR	Optimised Link State Routing
OTcl	Object Tool Command Language

VSHDRP	vehicle Second Heading Direction Routing Protocol
PD	Path Discovery
PDA	personal digital assistants
PGB	Preferred Group Broadcasting
QoS	Quality of Service
RF	radio frequency
ROMSGP	Receive on Most Stable Group-Path
RREP	Route Reply
RREQ	Route Request
RSU	Road Side Units
rtPS	time Polling Service
RVC	Road-to-Vehicle Communication
SADV	Static-node Assisted Adaptive Routing Protocol in Vehicular Networks
SCH	Safety Channel
SCMU	Speed Control and Monitor Unit
SHD	Second Heading Direction
SKVR	Scalable Knowledge-Based Routing
SN	source node
SNAR	Static Node Assisted Routing
SNR	Signal-to-Noise Ratio
SSA	Signal Stability-Based Adaptive
STAR	Source Tree Adaptive Routing

TORA	Temporarily Ordered Routing Algorithm
TTL	Time To Live
UGS	Unsolicited Grant Service
UL	UpLink
U.S. DOT	U.S. Department of Transportation
V2I	Vehicle to Infrastructure
V2V	Vehicles to vehicle
VADD	Vehicle -Assisted Data delivery Routing Protocol
VANET	Vehicular Ad hoc Network
VII	Vehicle Infrastructure Integration
VoIP	voice over IP
WAN	Wireless Local Area Network
WAVE	Wireless Access for Vehicular Environment
WBA	Wireless Broadband Access
WiMAX	Worldwide Interoperability for Microwave Access
WRP	Wireless Routing Protocol
WSN	WirelessSensorNetworks
ZRP	Zone Routing Protocol

Chapter 1

Introduction

Objectives:

- Give an introduction and the motivation of this research
 - List the research questions
 - Present the research methodology
 - Give the thesis structure
-

1.1 Research Motivations

Traffic congestion caused by vehicles crashes is considered to be an issue of great importance on the roads. Therefore safety applications are the focus of most researchers working in the area of VANET systems; as a consequence increasing the efficiency of these applications has a vital impact on their contribution to limiting the number of fatalities and providing safer, cleaner and more comfortable traveling on roads. Vehicle drivers have no ability to predict the conditions on the road ahead of [9]; however, with the aid of sensors, computer equipment and wireless communication devices, with a combination of technologically equipped devices, it is possible to provide methods by which vehicles on the roads can foresee the speed of other vehicles and assess possible risk. Through use of such methods, warning messages could be sent periodically to predict vehicle speed in order to eliminate the occurrence of accidents [10].

The unique characteristics of VANET are the high mobility and rapidly changing network topology caused by the high travelling speed of the nodes, the constrained pattern due to the restricted roads, limitations of bandwidth due to the absence of a central coordinator that controls and manages communications between nodes, disconnection problems owing to the frequent fragmentation in the networks and signal fading, caused by objects that form obstacles between the communicating nodes.

Consequently, the main challenge facing VANET is to decide upon the routing protocol that should be used to control the process of forwarding packets through nodes on the network, determining how to select the next-hop node to use to forward packets to their final destination, particularly in a sparse environment, depending on the presence of the unique characteristics of VANET. Therefore providing a robust

routing protocol is considered to be the most crucial solution in VANET [11]. This involves using new parameters to take the decisions regarding selecting the next-hop node to enhance the efficiency of the routing process and increase performance. As a result it is anticipated that designing an efficient routing protocol will aid in accomplishing the task of delivering packets to their destinations in VANET via a more realistic method, which promises to apply road safety efficiently.

VANET is a specific circumstance of MANET; many routing techniques have been designed in MANET to tackle the limitations with transmission packet delivery delay, involving packets being dropped, wasting bandwidth, mobility and security. These techniques could not be appended to VANET owing to its particular characteristics, such as the restricted mobility pattern.

1.2 Problem Description and Routing Solution

Many routing protocols have been proposed for vehicular ad hoc networks, in an attempt to deliver packets to their desired destination efficiently [12, 13, 14]. The process of how to find the most robust route to the desired destination varies depending on the mechanism used in that protocol. In the challenge of designing an efficient routing protocol in VANET, keeping up-to-date information about other surrounding nodes is considered to be a vital process, either by flooding control packets, as in topology-based routing protocol or by exchanging HELLO beacon message as in a position-based routing protocol, which is the concern of the work in this thesis. The number of HELLO beacon messages exchanged should be maintained at a minimum, in order to avoid overburdening the network with a high number of HELLO messages; some of which may be unnecessary or redundant. This will lead to many problems in performance in terms of overheads, conflict and collision.

As mentioned before, VANET is a core research topic for many researchers and developers, especially when seeking to design an efficient routing protocol to deliver messages to the desired destination with a minimum time delay, and with a good level of reliability in connected environments [15, 12].

The main goal of this research is to develop a robust routing protocol that focuses on promising to increase the system's connectivity, reliability and bandwidth consumption and to reduce the probability of packet dropping in sparse environments, leading towards an efficient mechanism, helping to increase road safety.

The critical function of the routing protocol is how to find a reasonable data packet route via which to send the packet from the source node to the destination node. This function is rooted principally in the process of selecting the next-hop node, which in turn will perform in a particular way depending on the knowledge of the surrounding neighbour nodes. The process of selecting the next-hop node has an impact on the connectivity and reliability of the network, and selecting the proper next-hop node will increase the possibility of obtaining a high network performance.

The number of nodes in the network has a crucial impact on network performance, when a high number of nodes are available in the network, which will increase the probability of selecting the most proper node to forward the packet to its desired destination, and in a case where a link breakage has occurred, an alternative node can be selected immediately with a minimum time delay; as a result, the packet delivery ratio will increase when increasing the number of nodes in the network.

However, the low number of nodes in the network can affect negatively on the

network's performance in terms of delivering the packet to its desired destination, which will lead to the creation of gaps between nodes (disconnected area), and as a result will prevent the packets being forwarded to their destination. Many routing protocols for sparse environments in VANET [12, 14, 16] have made proposals for increased efficiency of performance; these protocols have followed different approaches, in order to avoid dropping the packets when a disconnected area is created. The carry-and-forward strategy was utilised by the majority of those routing protocols, which state that the forwarding node will store the packet when no node is available to act as a next-hop node. In some situations the number of nodes may be high, but the distribution of these nodes may still lead to the existence of a disconnected area. For example traffic lights at intersections that are used to organise the traffic in the road are one of the factors that cause gaps between nodes.

1.3 Aims of the Research

According to discussion mentioned above, the aim of the work in this thesis is to introduce a solution to the following problems:

- Increasing the packet delivery ratio, by increasing the stability of the link route packet and decreasing the generated overhead, by reducing the retransmission of the packet caused by drops occurring.
- Increasing the system's reliability and connectivity, by decreasing the link breakage between two communicating nodes and provide the forwarding node with up-to-date information about the circumstances of neighbouring nodes to be utilised in the process of selecting the next-hop node.
- Reducing the overhead that results from sending beacon messages between

nodes by finding a new technique to organise the broadcasting of those messages in the network, which promise not to be unnecessary.

- Building architecture that supports the broadcasting of the HELLO beacon message adaptively in VANET.

1.4 Contributions

The main contributions that have been performed in this thesis are illustrated by four key ideas; these contributions are as follows:

- Second Heading Direction in VANET: A novel aspect of exploiting the second Heading Direction (SHD) as a parameter in VANET has been introduced, this new parameter has been utilised in the process of selecting the next-hop node to forward the packet to its desired destination. Each vehicle on the road can obtain the SHD through the navigation system (NS) and the global position system (GPS). The SHD can be provided by each vehicle on the road to its surrounding neighbour node through broadcasting it using the HELLO beacon message.
- Vehicle Second Heading Direction Routing Protocol (VSHDRP): A novel routing protocol has been designed for the sparse environment in VANET, this protocol comprises two modes; the first one (VSHDRP1) is convenient for straight road in highway conditions when the location of destination node is known, while the second mode (VSHDRP2) is more appropriate for intersection/roundabout conditions when the location of the destination node is unknown. This protocol falls under the category of position-based routing protocol for the delay tolerant network type. The mechanism of this protocol works based on the filtration process for the next-hop node, which is based on

four factors; vehicle position, current direction, SHD and speed.

- Formal specifications using CCA: A Formal specification has been performed for VSHDRP using the Calculus of Context-aware Ambients (CCA), in order to verify the behaviour of the vehicles in forwarding packets from the source to the destination and the flaw of these packets based on the proposed routing protocol VSHDRP.
- On Board Unit Architecture for Adaptive HELLO Message: An on board unit architecture for broadcasting the HELLO beacon message adaptively in VANET is based on the circumstances of nodes in the network such as the direction and speed; the architecture has also been designed based on the aspects of a context-aware system.
- Sending HELLO Beacon Message Adaptively in VANET: A novel technique for broadcasting HELLO beacon message adaptively in VANET has been introduced, this process of broadcasting the HELLO beacon message has been organised based on the vehicle circumstances on the road; rather than broadcasting these periodically, circumstances, such as the vehicle's direction and speed and any change to these circumstances will lead to a change in network topology. Therefore if there is no change in the network topology, sending the HELLO message periodically will produce a redundant and unnecessary message in the network, which will lead to an increase in the overhead, conflict and collision in the network.

1.5 Thesis Outlines

The following describes the outline of the remainder of this thesis.

Chapter 2

Presents an overview of VANET; describing the communications types, its characteristics and how these are related to MANET characteristics. Furthermore this chapter illustrates the challenges and applications. In addition it introduces the background to the routing protocols and provides a descriptive classification of vehicular ad hoc networks, some examples of routing protocols in each category have been described.

Chapter 3

Provides Preliminaries of the routing management in VANET, which include; the methodology, the background to the context-aware system, the key ideas relating to adaptive HELLO beacon messages in VANET and the key idea behind vehicle's second heading direction routing protocols in two modes (the straight highway mode and the roundabout/intersection mode). Moreover this chapter describes the system model and illustrates the assumptions made when designing the new routing protocol mechanism.

Furthermore this chapter introduces an overview of the formal method that has been used to build the formal specification and verification of the new protocol using the Calculus of Context-aware Ambient (CCA), and provides an overview of the Network Simulator version-2 (NS2) used to evaluate the new routing protocol.

Chapter 4

Presents the idea behind the adaptive HELLO beacon message in VANET; this describes the situation when broadcasting the HELLO beacon message adaptively. Moreover the on board unit architecture for adaptive HELLO messages (which was designed based on the aspect of the context aware system) is divided into three

phases based on the context-aware system. The three phases are described in detail to show the roles and functions of each phase in the architecture.

Chapter 5

Introduces the design and development of the first mode of the new routing protocol VSHDRP1; it includes the description of the new protocol mechanism, the formal specifications and validation of the protocol based on the Calculus of Context-aware Ambient (CCA). Furthermore it presents the evaluation of the protocol based on the Network Simulator version-2 (NS2).

Chapter 6

Demonstrates the second mode of new protocol VSHDRP2, it comprises a comprehensive description of the protocol mechanism. In addition it presents a formal specification of the new protocol, using the Calculus of Context-aware Ambient (CCA) for the purpose of validation, and to examine the system's behaviour.

Chapter 7

Demonstrates a case study and the considerable possibilities for the application of the new routing protocol in its two modes. This comprises an ambulance warning system on the highway and roundabout scenarios.

Chapter 8

Concludes the work presented throughout the thesis, providing some suggestions for future work.

Chapter 2

Overview of Routing Protocols in VANET

Objectives:

- Present the basic concept of MANET
 - Provide an overview of VANET
 - Provide an overview of routing protocol in VANET
 - Present a classification of routing protocol in VANET
-

2.1 Introduction and Motivation

In recent years, the number of vehicles has been increasing rapidly and continuously on the roads, and as a result of the road saturation and lack of attention by drivers, it is hard to keep safe distances and restricted speed between vehicles, which has led to an increase in the challenges and dangers faced by drivers. The efforts of automobile manufacturers with national government agencies have been gathered to develop solutions aimed at assisting in enhancing driver behaviour on the roads by predicting accidents or avoiding bad traffic areas. This combined effort has resulted in the employment of a novel type of wireless access called Wireless Access for Vehicular Environment (WAVE), which is dedicated to vehicle-to-vehicle and vehicle-to-roadside communications. While the main goal has clearly been to improve vehicular traffic safety issues, it has introduced traffic management solutions in addition to the on-board entertainment and information applications, such as internet access and gaming [17].

A VANET enables vehicles to create a self-organised wireless network between them on an as-needed basis. To establish a successful communication, an efficient path must be initiated to deliver the packet to its destination; this can be accomplished by designing an efficient routing protocol. To contribute in a VANET, transceivers and computerised control modules need to exist in the vehicles that provide them with the ability to communicate as network nodes. The range of each vehicle's wireless network may not exceed more than a few hundred metres; thus satisfying end-to-end communication over many miles needs messages to trip through several intermediate nodes. It is not vital to have a network infrastructure to create a VANET, although Road Side Units (RSU) may need to use a fixed network. These roadside units extend the use of many services for vehicular networks. VANETs are

a subset of mobile ad hoc networks (MANETs) [18]. Mobility is the main characteristic of most nodes in VANET; however, because of the constrained mobility pattern, vehicle movement is generally restricted to a plan of roadways; they have a distinct controlled mobility pattern that is subject to vehicular traffic regulations. In urban areas, high buildings and other obstacles between roads act as gaps to radio communication; therefore it is necessary to communication over roads. In general, the speed of vehicles is higher than that in other MANET nodes. Vehicles moving in the same direction and generally having the same speed can maintain a fixed gap between them; for that reason they keep contact with one another for much longer periods of time than with vehicles moving in different directions [19].

A routing protocol governs the way of communication between two nodes to exchange information; it determines the method of setting up a route, makes the decision about packet dissemination, and takes action in maintaining the route or recovering from routing failure. Unfortunately, although numerous routing protocols have been designed in MANETs, many of these protocols are not suited for application on VANETs. VANETs represent a main challenge class of MANETs. A VANET acts as a decentralised; self-organizing communication network, and is therefore considered by the mobility of the node and constrained by the restricted mobility patterns. Appendix A provide a historical background in MANET and a general overview in VANET.

2.2 Why MANET routing protocols are not suitable for VANET scenarios

Here we will show how the routing protocols in MANET is not suitable to be applied in VANET, owing to the difference that exist between the two types, reasons for

that can be summarised as follows [6]:

- *Scalability*: a limited number of nodes (1-2 hundreds) can be served by the routing protocols designed for MANET; these protocols compute the path used by a mechanism, which is considered to be very costly for a widely distributed network like VANET; it is not feasible to store routes to other nodes in the network, as in reactive and proactive routing protocols.
- *Full Connectivity*: in VANET, when the sending node transmits the packet to its destination, a chance of delay is possible with the Delay Tolerant Network (DTN), owing to the changing of vehicle position (high mobility); in other words, the path to the destination is not specified.
- *Mobility Pattern*: in VANET, the sending node has an idea about the mobility pattern which is represented by the restricted vehicle's movement by the road topology, speed limit and traffic signals. This consideration does not exist in MANET; instead it assumes an arbitrary pattern which is not efficient for mobility patterns as in VANET.
- *Disconnective path*: in VANET, the link breakage can be avoided with the aid of the mobility pattern knowledge for neighbouring nodes. In contrast, this information is not available for the MANET routing protocols; instead, MANET uses either the periodic messages or path creation periodically to conduct the problem of path breakages.
- *Flooding Operation*: the flooding operation is considered to be the basis operation in MANET, as in reactive routing protocols where the route to the destination is specified by using the flooding operation. However, in proactive routing protocols, messages are sent periodically; as a result of using the flooding operation, bandwidth is greatly wasted in this operation and reduces

the network performance, especially in large networks with a huge number of nodes, as in VANET.

- *Non-local Function*: the distributed nature of MANET requires that all nodes participate in the operation of routing path establishment and maintenances, to create a routing table in proactive routing protocols and contribute in performing the primary flooding needs. In contrast, VANET requires the localization routing solution to collect information from their neighbour nodes to deal with overheads and scalability, and to gain flexibility regarding the network conditions.
- *Using Supported Knowledge*: on-board units have the ability to provide vehicles with useful information that improves the performance of the routing protocols, such as the predicted path, velocity, direction, and road topology in digital maps. However, routing protocols in MANET are not supported with this information.

2.3 Classification of Routing Protocols in VANET

Traditional topology-based routing protocols face many obstacles in VANET; reactive routing schemes not suitable for promising a whole path and proactive routing protocols suffer from frequent changes in topology. However, position-based routing takes the position information into consideration in making routing decisions. In terms of extending the route lifetime, another routing group tends to use the direction and speed information of the nodes; this approach is more suitable for connected VANETs. In other words, it is not appropriate for sparse areas (partially connected network) [5].

Routing protocols in VANET can be classified into two major categories: topology-based routing protocols and geographic-based routing protocols, as shown in Figure 2.1. In topology-based routing, packets are forwarded using available knowledge about links that connect nodes in the network. However, geographic routing utilises the location information of its neighbour's nodes to complete the process of packet forwarding.

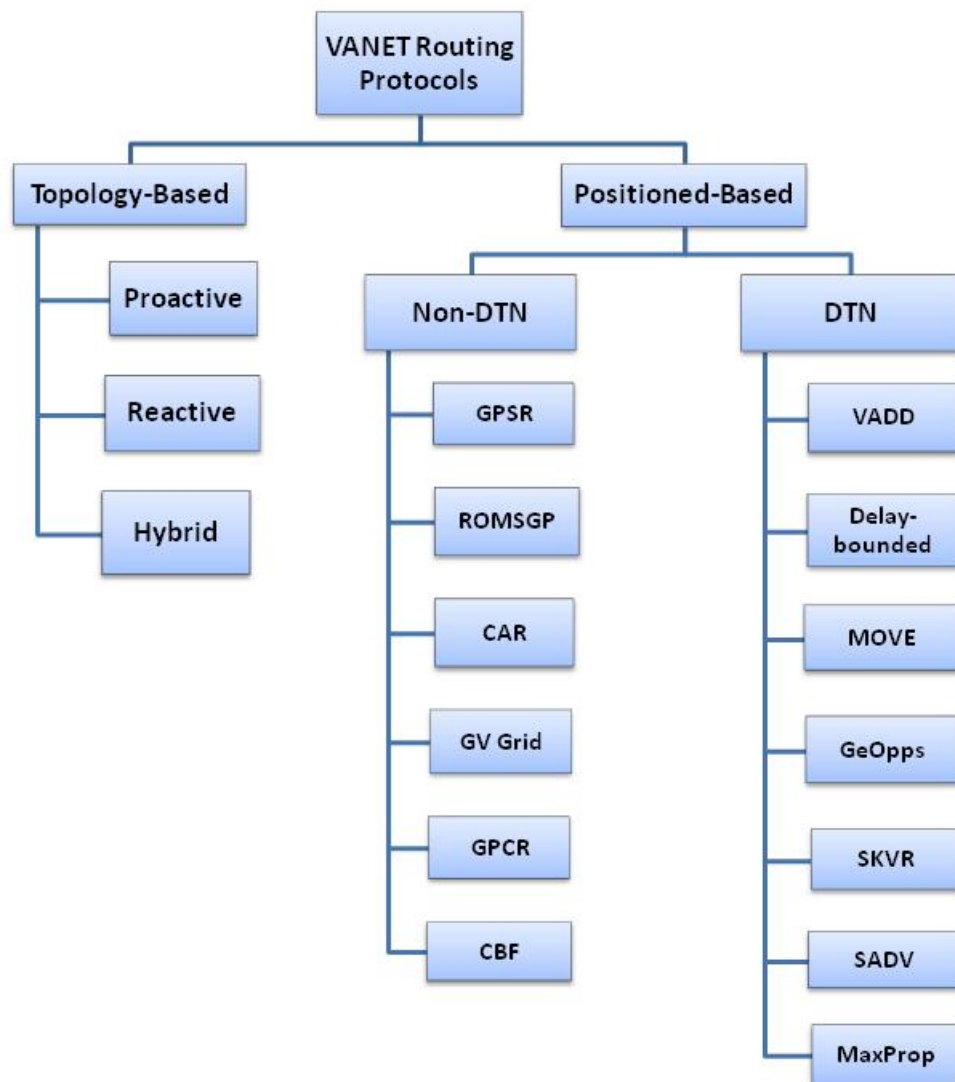


Figure 2.1: Classification of various routing protocol in VANET

2.3.1 Topology-Based Routing Protocols

This type of routing protocol is considered to be a conventional routing protocol. The link's information is the basis on the process of taking the decision to forward the packet; it can be generally classified into different categories based on routing strategy; proactive routing protocols (periodic), reactive (on-demand) and hybrid [20, 21, 22]:

2.3.1.1 Proactive Routing Protocols (periodic)

The proactive routing protocol is also called the table-driven routing protocol since each node maintains one or more tables to store the network topology and routing information. A periodic update is performed for exchanging this information by flooding the routing information in the whole network. The method of detecting and updating the routing information and the type of information maintained in the routing tables represent the difference between different proactive routing protocols. Distance vector and link-state routing protocols are types of this category. The classical Bellman-Ford routing algorithm [23, 24] is the basis of distance vector protocols, in which a list of all destinations and number of hops to each destination are maintained in each node. Accordingly, the performance of these protocols in a network containing mobile nodes is not acceptable because of the slow convergence and count-to-infinity problem (if a certain kind of link failure occurs in a routed network, the result is that the algorithm tries to count the shortest paths to infinity).

Thus, a number of proposed protocols have been introduced to improve the distance vector algorithm to be more fitted on mobile ad hoc network. Examples of such protocols are Destination Sequence Distance Vector (DSDV) routing protocol [25] and Wireless Routing Protocol (WRP) [26, 27]. In addition, different routing

protocols are proposed based on link state algorithms such as Optimised Link State Routing (OLSR) protocol [28], Fisheye State Routing (FSR) protocol [29, 30], Source Tree Adaptive Routing (STAR) protocol [31], and Global State Routing (GSR) protocol [32].

2.3.1.2 Reactive Routing Protocols (On-Demand)

This category of routing protocols is also called on-demand routing protocols. Dynamic topology is considered to be the major feature of mobile ad hoc networks, and especially for vehicle ad hoc networks; therefore, regular updates of the global topology information are essential at each node in order to chase the topology changes. This, however, consumes extensive bandwidth. To complicate matters further, the expiration of the updated received routing information before this information is sometimes necessary. In this situation, waste in a bandwidth may occur. The concept of reactive or on-demand routing protocol has been proposed by Johnson [33] in order to reduce the unnecessary routing information updates and the amount of bandwidth consumed.

Many protocols of this type have been proposed. The most typical reactive routing protocols are the Dynamic Source Routing (DSR) [34] [35], Ad hoc On-demand Distance Vector (AODV) [36], Temporarily Ordered Routing Algorithm (TORA) [37, 38], Associativity-Based Routing (ABR) [39, 40], and Signal Stability-Based Adaptive (SSA) routing [41].

Unlike proactive routing protocols, as a replacement for keeping the network topology information and route to each destination of the network, On-demand routing protocols [42] establish the necessary routes when required (on demand) by the source, by using the process of route discovery. Generally, when source S

needs a route, a Route Request (RREQ) packet is triggered and floods it into the network to build a route to the required destination, D . a Route Reply (RREP) packet is then sent back by D to S , when D receives the RREQ. If the route request has travelled through bi-directional links, RREP is sent using link reversal, or by piggybacking the route in a route reply packet via flooding. The main functions of a routing algorithm in on-demand routing protocols are route discovery and route maintenance.

2.3.1.3 Hybrid

The hybrid routing protocols are a class that uses a combination of both reactive and proactive routing protocols. In this type of protocol, the network scalability is increased by forming a near zone by the close nodes which work together to reduce the route discovery overheads by proactively maintaining routes to nearby nodes, and using a reactive strategy to determine routes to far away nodes. Most hybrid protocols proposed to date are zone-based, which means that the network is partitioned or seen as a number of zones by each node. In protocols belonging to this category, each given node partitions the area of the network into two distinct regions. The nodes in near distance from the node, or inside a particular geographical region, form the routing zone of the given node. In the routing zone, a proactive (table-driven) approach is used. An on-demand routing approach is used for nodes located in the area beyond the routing zone. The most typical hybrid routing protocols are Zone Routing Protocol (ZRP) [43] and Core Extraction Distributed Ad hoc Routing (CEDAR) algorithm [44]. The latter selects a minimum set of nodes as a core to perform QoS route computations.

2.3.2 Geographic-Based Routing Protocols (Position-Based)

This type of routing protocol is considered to be the most suitable type for VANET, depending upon the concept of using the position information by the source node to take the decision of selecting the best route to forward packets to its destination. In particular, the source node primarily takes the decision of forwarding the packet; first, the destination's position is maintained by the source node in the packet header, and second, it takes the one-hop neighbour's position into consideration. The information about this position is obtained by the beacons sent periodically with random jitter (to prevent collision); each node included in the radio range of another node is considered to be a neighbour to this node, and the geographical-based routing protocols work under the assumption that each node has information about its position, especially with the aid of the Global Position System (GPS) unit fitted on the on-board unit and navigation system [45], which provides the source node with sufficient knowledge about the position of the destination node.

Unlike proactive and reactive routing protocols which exchange link state and store information about established routing, geographic-based routing protocols are thus considered to be stronger and more suitable for a high mobility environment, as in VANET. In general, geographic routing protocols can be classified into two categories, as shown in figure 2.1: Delay Tolerant Network (DTN) and non-DTN.

The non-DTN types of geographic routing protocols do not take disconnectivity into consideration and are only suitable for high density environment, while the geographic routing protocol in DTN types take the disconnectivity in its account. Hybrid types of geographic routing protocols combine the non-DTN and DTN routing protocols to exploit partial network connectivity.

2.3.2.1 Non-DTN Geographic Routing Protocols

The non-DTN types of geographic routing protocols do not take the disconnectivity into consideration; it works under the assumption that there are always a sufficient number of vehicles to make a pair communication; therefore, this type of protocols is suitable for high density environments only.

2.3.2.1.1 Greedy Perimeter Stateless Routing (GPSR) The Greedy Perimeter Stateless Routing (GPSR) [46] is a routing protocol that decides the forwarding decision based on exploiting the positions of mobile node and the destination location of the packet. In GPSR, the positions of neighbours are maintained by the intermediate node without keeping routing metrics, which makes the protocol stateless. GPSR comprises two modes: greedy forwarding mode and perimeter mode.

The greedy forwarding approach can be performed when the packet received by an intermediate node is forwarded to the neighbour node that has the closest location to the destination; in case there is no neighbour node available closer to the destination than the intermediate node, this intermediate node is called the local maximum. At this stage, the role of the greedy mode is finished and the routing protocol switches to the other mode, which is the perimeter mode used to maintain this block end. The concept of the perimeter mode in GPSR is to utilise the rule of right-hand with the starting vector constraint for the purpose of forwarding the packet.

When the process of switching the packet to the perimeter mode at an intermediate node x is performed, a virtual vector is drawn from x to destination node D , using this intermediate node. After that, the packet is forwarded to the first edge counter clockwise from the vector by node x (an edge here represents a bi-direction

feasible transmission pair between two mobile nodes). Consequently, the next-hop in GPSR is found by the right-hand rule (the next forwarding edge should be the first edge counter clockwise from the previous edge without crossing the starting vector). The packet is switched back to greedy forwarding mode in the process of forwarding the packet to a node that is closer to D ; otherwise it will be dropped when a loop occurs.

The perimeter mode of GPSR must be applied on a planar graph, or the crosslink may cause routing loops. GPSR introduces two schemes to build a planar graph. However, issues such as obstacles and asymmetric radio range prevent planar graphs from being created correctly.

2.3.2.1.2 Receive on Most Stable Group-Path (ROMSGP) The ROMSGP was designed to raise the level of routing reliability by increasing the stability of the system; this can be achieved by focusing on the connectivity factor between nodes over the whole network. This protocol uses an approach that groups nodes into four groups according to their dot product of velocity vector and unit vector (VA.SN); in other words, by their direction, vehicles included in the same group are seen to make a more stable path. That approach helps the node to decide which path is more stable. ROMSGP uses the link expiration time (LET); the path with the longest LET is considered to be more stable than the others [47].

2.3.2.1.3 Connectivity-Aware Routing (CAR) One of the essential features which must be available in the routing protocol for VANET is its ability to deal with the continuous and rapid change in the network topology; a connectivity-aware routing protocol has been initiated to support the vehicular network in both city and highway environments. The key function of the CAR protocol is to determine the connecting path between source and destination nodes, in addition to defining the

position of the destination node. It uses a new aspect called *Guard* which makes it possible to track the current position of a destination.

The CAR protocol comprises four parts: destination location and path discovery, data packet forwarding along the found path, maintenance path using guards and recovery of error. Information about node direction and speed is added by all nodes periodically in the HELLO beacons; each node adds in its neighbours table the sender of the HELLO beacon when it receives it, and setting the entry expiration time in the table of the neighbours by estimating its velocity vector and that for its neighbours also. Furthermore, an update to the entry occurs, either when the gap between the position of the current node that had been estimated at the beginning of the entry and their neighbour positions tends to be greater than 80 percent, or after two HELLO intervals. Exploiting the velocity vector information leads to an increase in the probability of making the process of sending the HELLO beacon more adaptive. This can be used to avoid the process of sending a HELLO beacon in a periodic time interval, which is considered to be a negative aspect, as a result of the increase in bandwidth consumption, data packet delay and network congestion.

In the CAR protocol, an adaptive mechanism is used by sending HELLO beacons in changeable intervals based on the number of nearby neighbours that have been registered. In other words, the HELLO beacons will be increased frequently if the node has a few neighbours and vice versa. At the same time, it could be said that traffic conditions play an important role in determining the stability of the beaconing rate; this adaptive mechanism decreases the overhead efficiently without any effect on the routing techniques. Consequentially, the CAR protocol uses the aspect of guards with its two types: standing guards and travelling guards; it is represented

by an entry in the HELLO message; the Guard entry contains an id, TTL (time to alive), guarded position and radius. Information can be added to the packet and the node can filter packets if the guard is available in the node. The standard guard depicts information about an impermanent status that is attached to geographical vicinity; the node that receives the guard records it in its guard table and decrements the TTL each time the HELLO beacon containing the guard is retransmitted. The guard is removed when the TTL reaches zero, while the travelling guard uses the velocity vector in addition to the position and radius; the reaching time for the guard is recorded when the node receives it. For the destination location discovery, CAR limited the redundant transmissions to decrease the control message by using Preferred Group Broadcasting (PGB) in broadcasting mode to find the path to destination; the source nodes then records its id and the velocity vector keeps listening after sending the broadcast packet.

If the packet is not forwarded to another possible destination, the source node repeats the forwarding after a predefined time until the packet can find the next possible destination. Each node receives the packet and records its *id* in the table of received path discovery; an anchor point containing the coordinator and velocity vector of the current and previous node is added to the packet of broadcast. If the node velocity vector direction is not parallel to the previous one, the destination node can thus know the complete path to the source node. The decision of taking a certain route depends on the received Path Discovery (PD); the destination can reply simultaneously or wait for other PDs to select the appropriate one in terms of connectivity and delay. Advance Greedy Forwarding (AGF) is used to reply to the source; after that, the source uses the greedy forwarding method by exploiting the anchor point rather than the geographical information. If the path is invalid, maintenance for this path is activated by using the guards, and any changing in the

source or destination direction leads to a disconnection in the path. Therefore the importance of the guards emerges by using the information from the new and old velocity vector to update the information on the position of the destination [13].

2.3.2.1.4 GV Grid In the *GV Grid* [18], the goal is to obtain an on-demand route with high quality, and to elongate the pair connection lifetime starting from a fixed source to an exact region, with the help of external equipment, such as digital maps and GPS systems to provide position and direction information. The *GV Grid* is a positioned-based protocol based on dividing the broadcast area into grids which represent a set of equal size squares.

The concept of the *GV Grid* consists of two parts; the first grants best stability through finding the multi-hop path (network route), depending on the characteristics of nodes from the observation knowledge obtained through digital maps and GPS system to provide the positions and directions of vehicles, which in turn depends on the driving route characteristics. Secondly, each intermediate node in the driving route records information about it; in this way, a broken path can be redesigned without the need to send RREQ messages. The route discovery process aims to determine the longest lifetime based on the vehicle speed and driving route direction; in an attempt to find a path to designation D , node SDS specifies where the RREQ message needs to be sent, by determining the request zone which represents the smallest rectangle which includes source and destination nodes; after that, the S node forwards the RREQ message to an electing node from each neighbouring grid. In the same procedure, nodes in the neighbouring grids that receive the RREQ will select their neighbours.

2.3.2.1.5 Geographic Routing in City Scenarios (GPCR) The position-based routing protocol is considered to be an appropriate scheme for high mobility

environments such as highways; however, in urban areas, this approach faced some difficulties in radio propagation process owing to the nature of the city environment which contains high buildings. This problem can be solved through using a static map which provides the node with sufficient knowledge about the city's topology.

The position-based protocol depends on a greedy forwarding concept to forward packets to the next hop. This concept is based on forwarding the packet to a neighbour node which has a close position to the destination's geographical position. To accomplish this concept, the nodes that hold the packet have to be aware of its position, that of its direct neighbour and the destination node position. The own position can be gained via a GPS system, while the neighbour position can be obtained by the one hop periodic beacon; however, the destination position is gathered from the geocast application or location service. As a result of using local information in greedy forwarding, a local optimum may occur; to avoid that, a repair strategy is used which forwards the packet to node that is closer than the node holding the packet. Many other strategies, such as Greedy Stateless routing (GPSR) and face-2, were used.

The Greedy Perimeter Coordinator (GPCR) [48] is one of the position-based routing techniques which are based on the greedy forwarding and repair strategy; GPCR depends on street and junction points in making routing decisions, instead of using graph planarization. The repair strategy comprises two components: determine which street the packet takes on the junction, which is called a coordinator, and forward the packet to the next junction by using a special type of greedy forward. The problem here is determining which rule decides which node is on the junction, where A wants to send a packet to node C after the junction; in this case, the regular greedy forward cannot decide which node (intermediate) is a coordinator. Thus the

node between the source and destination will use a method which will be illustrated later, to establish whether it is a coordinator or not. Assume the node that wants to forward the packet is placed in the street instead of on the junction. In this case, the forwarded packet will be disseminated through the street to reach the next junction. To attain this assumption, it is necessary to send the packet to a node that extends the path between the source and destination. If no qualified (coordinator) node is found, the forwarder selects the node which has the longer distance from it. The coordinator node needs to decide which street is the selected node to forward the packet through it; this can be accomplished by using the right hand rule.

The main obstruction is to determine the coordinator not including outer information; this can be achieved by one of two alternatives: first, by periodically sending beacon messages which include the position of the node and that of its neighbour; according to that strategy, each node has sufficient information about its position, its neighbour's position and the position of the neighbours of each neighbour nodes. A node can be a coordinator if its neighbour list contains two neighbours located in the transmission range of each other, without being a neighbour to each other. The second alternative shows how to avoid using a beacon message by calculating the correlation coefficient according to its neighbour position.

2.3.2.1.6 Contention-Based Forwarding (CBF) Contention-Based Forwarding (CBF) is another geographic routing protocol designed by authors in [49]. Unlike other geographical routing protocols in VANET, CBF avoids using periodic beacon messages; furthermore, it works under the assumption that each node has sufficient knowledge about its geographic position.

This protocol sends data packets to all its first neighbours; then those neighbours take their decision to forward the data packet. The data packet for a CBF holds the last node forwarder's position (last-hop), the final destination ID and position and the ID of the packet. A distributed timer-based contention process which was initiated by the received node that is not the final destination provides opportunities to the most suitable nodes to forward the packet, while preventing other nodes from completing the forwarding process.

Receivers of the broadcast data would compare their distance to the destination with the distance of the last hop to the destination. The bigger the difference, the larger the progress and the shorter the timer. CBF shows an improvement in the packet delivery ratio (PDR) when it is compared with GPSR; with the increase in beacon interval, the PDR will decrease in GPSR. In addition, CBF avoids using constant overheads, as in GPSR, which leads to a reduction in the load on the wireless medium communication.

2.3.2.2 DTN-Geographic Routing Protocols

Delay-tolerant networks, also called distribution tolerant networks, are considered to be an effective solution for a low density of vehicles environment, such as in rural highway conditions, night conditions in cities and even in urban environments with sparse sub networks scenarios, owing to the high mobility and insufficient number of vehicles in a certain situation that acts as an obstacle, which prevents the establishing of an end-to-end communication between vehicles, which leads to frequent disconnections.

A carry-and-forward strategy has been devised to fill in the radio gaps between vehicles, by buffering the packet in case there is no appropriate node to forward the

packet and carry the packet to another area until an opportunity arises to communicate with an appropriate node to forward that packet; as a result, capability to exchange information between vehicles is achieved by initiating a robust connection with high reliability and connectivity. A set of routing protocols that rely on a carry-and-forwarding strategy are described in more detail below:

2.3.2.2.1 Vehicle -Assisted Data delivery Routing Protocol (VADD) Because of the high mobility and unreliable nature of VANET, applying multi-hop data delivery tends to be more difficult. The VADD protocol depends on the store-and-forward aspect to realise the process of sending packet from the moving source to the fixed destination, taking the decision of electing the next hop node regarding the mobility of vehicles which are restricted by traffic patterns and road layouts.

VANET can be beneficial for many applications: for instance, road safety in terms of providing attention to a traffic jam, disseminating emergency warning messages and in case of vehicle needs to make a query from a fixed station in an uncovered internet area, exploiting the high mobility of vehicles to relate them with limited short hops. Authors in [12] designed VADD to tackle the dilemma of data delivery in VANET with the assumption that a digital map is used to provide information such as vehicle speed, intersection traffic signals, vehicle density, when a requirement to certain information appears from fixed station information. VADD introduce a reasonable solution to deal with the problem of data delivery, specifically in packet delivery ratio, data packet latency and overhead.

Although Greedy Perimeter Stateless Routing (GPSR) is considered to be one of the techniques based on the concept of geographic forwarding, which is quite efficient in delivering data, nevertheless it is inappropriate for low density vehicle

networks. In this case, the vehicle tends to keep using the wireless communication channel, or else forward the packet to a faster vehicle. Therefore, VADD will take three main principles into account: the first is to keep using the wireless channel if there is an opportunity for that; the second is to choose the higher speed in the road if carrying the packet is needed, and the last principle states that, as with the nature of MANET, VANET is a changeable environment, thus many selection processes are done during the dissemination process because it cannot be sure that the packet will be delivered through a predefined path. To gain the most efficient path to forward the packet, VADD depends on the coordinates of the vehicle that holds the packet, by switching between the three modes that the VDDA contain namely intersection, straightway and destination mode.

In intersection mode, based on the distance to destination, concerning the delay for packet delivery and packet delivery probability, the forwarding node can determine the priority of available streets and select the next intersection in the highest prioritised direction that depicts the target intersection. After that, it will attempt to elect a candidate node for that intersection, if it fails to find an appropriate candidate node, it will take the second best direction, select a target intersection in that direction and resume the strategy. If there is no suitable node, the forwarding node carries the packet. In an attempt to select the next-hop node by the forwarding node, the VADD protocol has several alternatives with different conditions. The first one is the Location First Probe VADD (L-VADD), which follows the same steps of the position-based greedy strategy; it selects the closest node to the target intersection without taking into account the travelling direction. This approach suffers from the loop that may occur, which has a negative impact on delivery ration. The second one is the Direction First Probe VADD (D-VADD); this approach depends on its work in selecting the node that has the same direction, and it assists

by eliminating the loop effect. Another approach that has been introduced is the Multi-Path Direction First is the Probe VADD (MD-VADD) which selects a multi path direction instead of one direction; however, this approach suffers from wasting bandwidth by using redundancy packets. Finally, a hybrid Probe VADD (H-VADD) has been introduced, which combine the using the L-VADD and D-VADD by taking the positive impact of each one. It focuses on using the L-VADD at the first stage; if a loop is detected, it switches to D-VADD. This approach gave better results than using L-VADD or D-VADD alone.

VDDA provides a delay model to evaluate the data delivery delay in roads by depicting the vehicle with a node while the road is by an edge. Its direction represents the traffic direction and the weight of the edge acts as the delay in the forwarding packet.

The intersection mode used in the VADD protocol is considered to be the most challenging situation; two main questions arise in this mode: by which road the packet needs to be forward and which the best carrier is. A sophisticated method is used to decide which direction road should be selected next; When generating a linear equation system nxn , n represents the number of roads with boundary intersections it solves by using the Gaussian elimination algorithm.

2.3.2.2.2 Delay-Bounded Routing in Vehicular Ad-hoc Networks The authors of Delay-bounded Routing in Vehicular Ad-hoc Networks [50] introduced two algorithms, D-Greedy and *D*-MinCost, in an attempt to deliver data from a moving node to a fixed infrastructure; the algorithm environment is a densely urban area, so the authors concentrated on achieving ideal bandwidth consumption in terms of reducing the number of transmitted messages by exploiting traffic in-

formation, such as traffic density and vehicle speed, to take the right decision on selecting the exact protocol and road segment. *D-Greedy* utilise the information of local traffic, while *D-MinCost* uses global information about the city; in order to obtain a reliable connection in a low density area, a store-and-forward technique is used. The main concept of this algorithm is its ability to alternate between two approaches, which can be used to achieve the goal of reducing transmitted messages; the first one is multi-hop forwarding, which is based on forwarding the packet to the next hop with closest location to the destination; the second approach is called data muling, which depends on the store-and-forward aspect.

The first algorithm is *D-greedy* which depends on local information to determine the shortest path which is considered to be the required path to the access point (*AP*); a neighbour list is created in each vehicle by sending a periodic beacon message. In addition, the id of the vehicle beacon contains the shortest path between the vehicle and access point (*dist To AP*) which is determined by using Dijkstra before the process of broadcasting. The strategy selection decision (multi-hop forwarding or data muling) to forward the message depends on a delay budget which is assigned to each edge according to its length; if the time of sending the message is still less than the threshold, a muling strategy is assigned; otherwise a switch is made to a multi-hop forwarding strategy.

In contrast, the *D-MinCost* algorithm takes into account global information such as vehicle speed and density in each road segment; to obtain a reliable connection in a low density area, a store-and-forward technique is used. Two metrics are considered in selecting the best path with the suitable strategy (multi-hop or data muling); the cost (*C*) of edge in terms of number of the messages in that edge, and delay (*Del*) depicted by the time needed by the message to travel through the edge; as a result

the path will be chosen according to the minimum cost.

2.3.2.2.3 Motion Vector Routing Algorithm (MOVE) This algorithm [16] has been designed to deliver a message from a vehicle to a static destination (fixed station like roadside unit) in a sparse environment, where a few opportunities for routing options exist; the vehicle operates as a mobile intermediate that has discrete connectivity with other network nodes. It focuses on predicting which vehicle from its neighbouring vehicles will travel towards the fixed destination by utilising d from the knowledge of its neighbouring vehicles, such as velocity and trajectory. Furthermore, in this algorithm, the store-and-forward strategy is used to save and store data and hold it for a period of time when no vehicles are available to act as intermediate nodes. Because of the continuous topology changes and a rare occurrence of connection opportunities, it is important to exploit it efficiently, by exploring whether in such opportunity forwarding a message is anticipating in achieve the destination.

The MOVE algorithm assumes that each node in the network has sufficient information about its position and direction, in addition to knowing the location of the fixed destination D where the message needs to be sent to; from that information, the closest distance (d_C) between the current vehicle (C) and fixed destination (D) can be calculated by the current vehicle. The source node C sends a HELLO message in a periodic time, and the neighbour N node which receives this message will send back a RESPONSE message to inform the Node C of its existence. The current node C can determine the shortest distance d_N from the its neighbour N to the destination D , by having direction information from that neighbour which will inform it about its trajectory. The current vehicle C can take its routing decision based on the information of each vehicle's d_N , and rules about each vehicle's trajectory; for instance, when moving C from its trajectory, it is necessary to forward

it to its neighbour, who will move toward the destination.

Compared with the greedy position-based routing algorithm, the MOVE algorithm has a higher data delivery rate in sparse environments, in addition to using less buffer space; however, the performance of the greedy position-based routing algorithm appears to be better in the primary evaluation where vehicles always behave in the same way or have the same attitudes, like bus routes.

2.3.2.2.4 Scalable Knowledge-Based Routing (SKVR) In the Scalable Knowledge-Based Routing (SKVR) [51] algorithm, the predictable characteristics of public transport routes and schedules have been exploited; bus routes are particularly used. Under the SKVR assumption, the bus route has a begin point and finish point, rather than a looped route. It is travelling in a forward and backward way. In this algorithm, the network is divided into domains, and each domain represents a separate bus route.

The SKVR protocol has two levels of hierarchy: top and bottom levels; the top level represents the inter-domain level, which deals with situations that include source and destination node in different bus routes, whereas the bottom level is intra-domain routing, which has source and destination nodes in the same bus route. It is important to know that the route of public transport cannot be predicted absolutely, owing to the occurrence of traffic delay, mechanical problems, driver behaviour, route conditions and other causes that lead to changes in routes. For that reason, two types of knowledge have been used: static knowledge that is not affected by time and dynamic knowledge which tends to change sometimes. SKVR depends on static knowledge (bus routes and intersections) solely for inter-domain routing; however,

SKVR utilises static and dynamic knowledge together for intra-domain routing; in this case, the dynamic knowledge can be represented by the timetable schedules for the bus, which may tend to change sometimes.

For intra-domain routing, when there is a need for sending a message from a source node to a destination node that lay on the same bus route, only two directions are available for sending the message: forwards or backwards. For the purpose of determining the position of the destination node without the aid of the location information, each vehicle stores a list of other vehicles it has recorded that are moving in the same route, and this list needs to be cleaned at the end of each route. The direction of each vehicle must be transmitted over the route during its journey. If the destination is in this previous-contact list, then it must be in the backward direction, and thus the message's direction flag is marked accordingly. When vehicles along the same route encounter one another, depending on the information of the vehicles' direction, the node carrying the message makes the decision of whether to forward it or keep it in a buffer.

In this manner, messages are forwarded to vehicles moving over the route in the same direction of their direction flag until they reach their destination. Ultimately, each vehicle over the present route will attain the route end when travelling back over the route in the backward direction. This provides a guarantee, that if a disconnection occurs; the stored messages will be delivered when they cross paths with another vehicle in their route.

For inter-domain routing, a message is forwarded to a vehicle moving in a destination domain or to a vehicle that can communicate with that domain; when it reaches the domain of destination, the delivery operation can be completed by intra-

domain actions. However, if the contact list of sending vehicles does not include any vehicles in the destination's domain, copies of the message are disseminated to other vehicles in the contact list.

2.3.2.2.5 Static-node Assisted Adaptive Routing Protocol in Vehicular Networks (SADV) The purpose of designing the Static node Associated Adaptive Routing Protocol in Vehicular Network (SADV) [52] is to eliminate the delay of message delivery in sparse environments; the SADV protocol found that the VADD protocol has some limitations which are represented by the problem of instability increasing as the density of vehicles decreases. After the dependence of VADD on the information of probabilistic traffic density and unavailability of optimal path, static nodes can be used to treat this limitation at intersections to help in delivering packets; these static nodes have the ability to buffer messages until it can find an appropriate node travelling on an optimal path. Under the assumption that SADV has sufficient knowledge about its own position with the aid of a GPS system, and access to a digital map, the SADV protocol allows each node to calculate the time needed to deliver a message, which leads to this protocol becoming more adaptive to traffic density variations.

SADV introduces three modules: Static Node Assisted Routing (SNAR), Link Delay Update (LDU), and Multi- Path Data Dissemination (MPDD). SNAR selects the path with the shortest expected delay to make a message delivery operation, with the aid of static and dynamic nodes; this is accomplished in two modes: the first is an in-road mode which is activated while a message is being carried by a dynamic node in a road; in this mode, a greedy protocol is used to deliver the message to a static mode at an intersection. In this stage, the second mode (intersection mode) is activated.

In the intersection mode, the optimal next intersection for the message is calculated by the static node based on its delay matrix; this can be accomplished by saving the packet at a static node and forwarding it when a vehicle travelling toward the optimal next intersection is available; if more than one vehicle is travelling toward that target, the one closest to that target is selected. Buffer management is a vital concern in any store-and-forward communication, which prompted the authors in SADV to take it into consideration; this task is performed by forwarding the packet stored in the static node to a suitable available path to eliminate the stored messages. The strategy (least delay increased) is used to determine which message needs to be forwarded without increasing the delivery delay significantly.

As mentioned before, in SNAR an abstracted graph from the road map that has been weighted with delays of expected forwarding paths is used to determine the optimal paths. The delay matrix is kept dynamic by the LDU module; this task can be accomplished by calculating the delivery delay for message between static nodes. A timestamp is fixed in a message header when static node S_a receives a message, and the elapsed time is measured and timestamp is updated in the header of the message after the message reaches the next static node S_b , which can conclude the delay for the arriving messages from all incoming locations by maintaining the elapsed time in its buffer. This information can be disseminated periodically to all other neighbour static nodes.

2.3.2.2.6 Geographical Opportunistic Routing (GeOpps) For the purpose of forwarding a message in sparse networks, the carry-and-forward technique and opportunistic routing have been employed by the geographical opportunistic routing protocol [14]; it utilises the information provided by the GPS to provide each node

with sufficient information about its position, and exploits the knowledge of navigation systems (NS) to accomplish the process of packet routing to its destination efficiently, and to provide the fixed roadside unit locations.

In the process of sending the message from the initial source to its destination, the next hop in the GeoOpps can be selected by the intermediate nodes according to the following method. The future nearest points are calculated by each neighbour vehicle that tracks the route given by the navigation suggested; moreover, the amount of time required to reach that point(destination) is calculated by the utility function which is built into the navigation system.

The next hop vehicle is elected according to its estimated time of delivery for the packet; the vehicle calculates that the shortest estimated time will be the next hop node that carries the packet.

The authors of GeoOpps assume that some situation can occur; for instance, if a driver fails to track the route suggested by the NS in a certain point, in this case this vehicle needs to forward the message to another vehicle from its neighbours, and if the vehicle has stopped for any reason and stays in the stopped situation for a time, all messages held by this vehicle need to be forwarded to other vehicles.

In this routing protocol, some difficulties may arise in terms of calculating the time by the utility function, which depends on some factors considered by each type of navigation system, such as speed limit, traffic condition, road types, and average driver speed.

2.3.2.2.7 MaxProp The MaxProp protocol [53] employs the carry-and-forward technique and packet prioritization technique to leverage the level of message deliv-

ery in the network. This algorithm is investigated in a real network known as UMass DieselNet, which is implemented on buses; it permits each bus in the network to send its location and performance information to a wireless access point or to other buses in the network. It works under the assumption that each node has sufficient knowledge about its position by using a GPS system.

The operation of MaxProp can be divided into three phases: first is the neighbour discovery phase, where nodes discover each other before transferring any packet; the second phase is the data packet transfer stage, where data can be transferred between nodes in this stage, and the final phase is managing the storage, in which the buffer of local storage is managed by each node through selecting the packet to be deleted, based on a certain prioritization algorithm.

By default, the message will be carried by a node until the occurrence of the next neighbour discovery stage; a forwarding of the message will be performed at each time the node meets another node until the message timeout has elapsed. Confirmation of the delivered message is accomplished by an ACK in the data transfer stage, or a dropping occurs in the storage management stage because of a full buffer.

In the second stage, which is the data transfer stage, high-priority-first schemes specify the mechanism of data transfer, while in the storage management stage, messages are deleted according to the lowest-priority-first scheme.

The estimated cost is the basis of determining the priority of delivering messages buffered in a node to its destination. This estimation delivery cost is determined by allowing each node in MaxProp to maintain the probability of meeting every other node in the network. As each meeting occurs between nodes, an incremental averaging is used to modify this probability and these probabilities are exchanged

to between nodes at each meeting as well. As a result of all these values, the estimated cost of the delivery message for each possible path to that destination can be calculated by each node.

This is done by summing the probabilities that each connection on the path will not occur. The estimated cost for that message to reach its destination is the least costly path among all of those calculated. When the data transfer phase is established, an information exchange occurs in a predetermined order. First, the messages intended for the neighbour node are transferred. Secondly, an exchange of aforementioned meeting probability information between nodes is performed. Third, to provide safe deletion of the messages in the buffer, an exchange for delivery acknowledgments is accomplished. Fourth, messages with low hop count are transferred without taking the estimated delivery cost into consideration. Finally, other messages that have not yet been transmitted are disseminated according to their priority, depending on the estimated delivery cost.

Two factors (delivery cost and hop count) impact on prioritizing according to the following manner. When the message stored in the node buffer has a count hop less than the t threshold, hops are stored by their hop count (high priority for new message); however, messages with a larger hop count have priority by the criteria of the estimated delivery cost. The maintenance issues of each node's finite buffer are handled in the storage management stage. When the buffer is quite full, messages need to be deleted. First, messages that have received acknowledgments are deleted. Secondly, messages with a hop count greater than the t threshold are deleted, depending on the highest estimated delivery cost first. Finally, if the buffer remains full, this can be mitigated by deleting the messages with hop counts less than the t threshold hop, depending on the largest hop counts first.

MaxProp has introduced a mechanism about prioritising messages, buffer management and routing messages, depending on the probability of meeting nodes. However, this technique is not suitable for a large distributed network.

2.4 Why Focusing on DTN Networks

According to the classification of routing protocols that has been introduced in this chapter, many proposed routing protocols have been described to show how each type of routing protocol has participated to enhance the performance of the network. Due to the unique characteristics of VANET compared to MANET, the position-based routing protocol is considered to be more suitable for VANET than the topology-based routing protocols.

Consequently, most of the routing protocols that have been developed for VANET are assumed that there are a high number of nodes available in the network to act as intermediate nodes. However this assumption is considered to be unrealistic, because in many situations number of nodes will be very low either along the road or in some sections of the road, This will reduce the packet delivery ratio and increase dropping packets. As a result, this will degrade the network performance. Therefore the focus of this thesis is on the DTN network type rather than the non-DTN network type.

The requirements and properties of the DTN routing protocols are summarised in table 2.1 to show differences and similarities among those routing protocols, which will support the process of developing our proposed routing protocols.

Protocol	V2V	Position based	Sparse	Gready Forward	Carry Forward	map	Traffic data
VADD		x	x	x	x	x	x
Delay-Bounded		x	x	x	x		x
MOVE		x	x		x	x	
Geopps		x	x	x	x	x	
SKVR	x	x	x				
SADV		x	x	x	x	x	
Maxprop	x	x	x		x		

Table 2.1: Requirements of routing protocols in DTN VANET

2.5 Summary

In this chapter, a comprehensive classification has been presented to illustrate the type of routing protocol in VANET, which each category comprises a set of routing protocol, which have been discussed to show the main idea of each protocol and specifies the strangeness and weakness of each protocol.

Some characteristics like mobility are considered as a drawback in some of routing protocol especially those who are fall under the topology-based category and more suitable for MANET, however this characteristics are provides strength for other types of routing protocols (position-based) which is considered to be more proper for VANET.

The position-based routing protocols have shown a high performance especially for non-DTN networks, as the number of nodes in this type of network is assumed to be high. In contrast, The number of nodes in DTN networks is tend to be very low in some situations, which has an impact on the performance of the routing protocols. All that motivated our work to focus on DTN networks as it shows more challenges rather than non-DTN network.

Chapter 3

Preliminaries

Objectives:

- Defines the aim of the routing protocol and specifies the objectives that need to be achieved
 - Presents and describes the main aspects of the proposed routing protocol mechanism
 - Presents the methodology the research will follow to evaluate the proposed routing protocol
 - Presents an overview of the context-aware system
 - Presents an overview of CCA formalising method
 - Illustrates the value of the parameters used in the environments of the NS-2 simulator
-

3.1 Research Introduction and Motivation

In the last few years, many researchers have concentrated their efforts on mitigating the routing management's issues in VANET, in order to enhance the performance of networks. Many routing protocols have been tried aiming to provide a better performance to exploit basic network services and information. However the use of such information can be beneficial when focused on a specific scenario, rather than being generally applied to address multiple situations. Owing to the unique characteristics of VANET, such as the high mobility and constraint patterns resulting from restricted roads, it is difficult to establish a route request and reply process, which can be used in a topology-based approach such as in the AODV routing protocol. Therefore position-based routing protocol are currently considered the best choice for VANET [6], such as the GPSR routing protocol; in this category each vehicle obtains knowledge about other nodes in the network, by broadcasting the HELLO beacon message to the surrounding neighbour nodes, which contains information about that node, such as the current position, direction and speed. Broadcasting the beacon HELLO message in regular time provides each node in the network with up to date information about the other nodes.

However, it results in a heavy load being generated [13, 49], which in turn leads to an increasing of the overheads, collision and contention, especially in high density networks. To overcome this drawback, a novel approach has been introduced in this thesis, which organises the operation of broadcasting the HELLO beacon message in VANET, this approach takes into account specific parameters. In other words each node in the network broadcasts the HELLO beacon message based on its circumstances, according to these parameters (broadcasting HELLO beacon message adaptively). In addition a novel context aware architecture, based on the adaptive

HELLO beacon message has been developed, which shows how this operation is related to the OBU.

In addition, authors [48, 47, 13, 46] have developed approaches based on exploiting geographical information to improve network performance in high density network, without taking into account the networks in low density environments. This type of network is called the Delay Tolerant Network (DTN), and suffers from having disconnected areas, due to the unavailability of intermediate nodes allowing the forwarding of the packet to its destination. Proposed routing protocols in [12, 50, 16, 14, 52, 51] have been developed to mitigate this dilemma, by exploiting the aspects of the carry-and-forward strategy, which state that if no neighbour nodes can be found to forward the packet, then the node that holds the packet can store the packet in its buffer and forward it when it reaches a new area where a new neighbouring node can be found.

In reality, the node that has been chosen as a next-hope node is considered to be inappropriate in some circumstances, for example if the node is chosen according to its location being closer to the destination than the holder node, without taking into account the direction and speed of this node, then there is a possibility that this node is not the right choice. For that reason, in this thesis a novel routing mechanism for DTN in VANET has been introduced, in order to resolve the dilemmas that occur in sparse environments. This proposed protocol is called the Vehicle Second Heading Direction Routing Protocol (VSHDRP).

The work carried out in this thesis falls under the DTN routing protocol category, according to the discussion mentioned in the previous chapter. The main drawback in all the routing protocols proposed in DTN is that, there is no guarantee that

the next-hop node will continue on the same route towards the destination, which results in dropping the packet after a specific period of time, leading to a decrease in the packet delivery ratio and the need to re-transmit the dropped packet, which means a lot of delay and increased overheads. Therefore, this drawback is considered as decreasing the efficiency of the system, and is likely to be very costly.

3.2 Research Methodology

This thesis follows a research methodology as follows, in order to achieve the pre-determined objectives:

1. specifying the research problems, which can be accomplished by a concentrated literature survey.
2. finding and developing techniques to solve these problems.
3. improving and expanding the techniques applied to solutions.
4. evaluating and examining the research concept via formalising and simulation tools, then comparing the result with a standard benchmark.
5. providing a result analysis, in order to determine the strength of the concept and its weaknesses.
6. organising documentation and specifying the possible directions of future work.

The investigation in this thesis has been established using a two phase literature survey covering; VANET overview and routing protocol in VANET. In the first phase an overview was introduced in VANET, to give a general view of the VANET characteristics and applications and compare these with MANET, as VANET is considered as a subset of MANET [18].

The second phase discusses the routing protocols in VANET in depth, as our main objective in this thesis is to design a new routing protocol in VANET, to improve the system's performance, in terms of increasing the packet delivery ratio and increase the system's stability. Thus it illustrates a classification for routing protocols in VANET and discusses the drawbacks of each routing protocol identifying into what category each should fall.

3.3 The Key Concept of the Research

This thesis has introduced two key concepts in order to manage routing packets in VANET; these concepts can be illustrated as follows:

3.3.1 Adaptive HELLO Message in VANET

One of the key ideas that have been introduced in this thesis is to send a HELLO beacon message adaptively in VANET. The majority of the proposed routing protocols in VANET utilise the technique of broadcasting the HELLO beacon message periodically to each node in the network, for instance every 0.5 seconds, in order to inform the surrounding neighbouring nodes about its situation, the HELLO beacon message contains information about that node, such as position, direction and speed; in such situations other nodes can acquire sufficient knowledge about the other nodes in the network. In this way each node can decide which one is more appropriate to act as a next-hop node. However broadcasting the HELLO beacon message periodically can generate a huge traffic load, especially with a high density network, which in turn will affect the quality of communications between nodes in the network, by increasing rivalry and collision.

To overcome this drawback, a novel technique has been developed in this thesis;

this technique allows the HELLO beacon message to be sent adaptively according to specific circumstances relating to the nodes. As mentioned in chapter two, one of the characteristics of VANET is that it is restricted by a certain mobility pattern represented by the road; thus our proposed technique allows the possibility to broadcast the HELLO beacon message adaptively based on two conditions (speed and direction), in other words if the node (vehicle) changes its direction, then it will broadcast the HELLO beacon message; or when there is a significant change in speed, which may lead to change the network topology, then the node can then broadcast the HELLO beacon message. As a result that technique will reduce the number of HELLO beacon messages, by avoiding broadcasting these messages on occasions when there is no change in the vehicle's direction or speed.

Based on the technique of an adaptive HELLO beacon message, a novel architecture has been developed; to prove the effectiveness of this technique, this architecture has been built based on the concept of the context-aware system, showing the three main phases represented by; sensing, processing and reacting, it shows how the OBU can be used to sense the direction and speed; also, the method of processing it applied, in order to decide how, when and where to take the right decision. This novel technique and architecture will be introduced fully, and in detail in chapter four.

3.3.2 Vehicle Second Heading Direction Routing Protocol in VANET (VSHDRP)

The VSHDRP comprises two main modes; each of them represents a different case study. Chapter five introduced the first mode, represented by VSHDRP1, while chapter six discusses the second mode, denoted by VSHDRP2.

3.3.2.1 The principles of VSHDRP1

The mechanism of VSHDRP1 represents the first mode of the proposed routing protocol, which has been developed for the highway scenario. The idea of this protocol is represented by exploiting the knowledge of the Second Heading Direction (SHD). With the aid of an equipped GPS and NS, each vehicle can obtain knowledge about the next road direction the vehicle is intending to follow; this information can be sent by each node in the network to its immediate neighbours through the HELLO beacon message. In this mode the SHD is shown to have only two values; "*zero*" which means the vehicle will continue along the same road and will not take the next exit, while "*one*" represents that the vehicle is intending to change its direction by taking the next exit. Taking knowledge of SHD into account in selecting the next-hop node will improve the system's performance in terms of reducing the route breakages and increasing the system's stability and connectivity even in a high density network.

Generally, nodes in VANET have tended towards participating with, and communicating with each other in an attempt to transfer the packet to its desired destination. By acting as intermediate nodes, all nodes take responsibility for receiving and forwarding packets from and to another nodes in the network, this crucial collaboration pushes all the nodes in the network to share some information with each other, in order to guide other nodes towards the right decision, which will be achieved while processing the packet that needs to be forwarded toward its destination. In VSHDRP1 the SHD is one piece of information that needs to be shared between the nodes in the network. This information can be stored in the buffer of each received node.

3.3.2.2 The principles of VSHDRP2

The mechanism of VSHDRP2 represents the second mode of the proposed routing protocol, which has developed for the purposes of managing intersection and roundabout scenarios, when the destination node's location is not precisely given. In this mode the SHD for each node is taken into consideration in the process of selecting the next-hop node. However in this mode the forwarding node will divide its neighbouring nodes into four zones (at least) based on their SHD, and the SHD value will then be determined according to the angle of the road that the vehicle is intending to follow, in this manner the forwarder node can accrue knowledge about where each neighbouring node is intending to go after the next intersection or roundabout. In this situation more than one node can take the same direction, and because the forwarding node already knows where each vehicle is intending to go, it can select one node from each group to act as an intermediate node to transmit the packet in that direction, instead of broadcasting the packet to all neighbour nodes. The process of selecting which node is going to be an intermediate node in each group is based on an algorithm that takes the current direction, speed and node location into consideration. This technique can promise to deliver packets to destinations in all directions, especially if the location and direction of the destination node is unknown. Meanwhile, there is no need to broadcast the packet to all the neighbouring nodes, as this has an impact on reducing the overheads generated.

3.3.3 VSHDRP Model Definition

This section describes the definition of the system model for the proposed routing algorithm, determining the assumptions of the system model and demonstrating the network model, mobility model and traffic model. As VANET is considered to be one of the more vital applications on an ad hoc network [18], a peer to peer mode

can be utilised as a communication technique between vehicles; as the proposed routing protocol entitles VSHDRP, as has been the case during this thesis.

A comprehensive algorithm has been developed, in order to accomplish all the tasks and functions that the proposed routing protocol needs to perform in its two modes (VSHDRP1 and VSHDRP2), a filtration process has been initiated, in order to select an appropriate next-hop node, this filtration process comprises four stages, which have been set based on exploiting the knowledge of the nodes in the network (position, direction, SHD and speed). This information plays an important role in expressing the circumstances associated with each node; therefore taking this information into account when processing the packet can effectively influence the network metrics; for example delivery ratio, overhead and time delay. This algorithm can be applied in each network in VANET, in order to improve the routing performance.

3.3.3.1 System Model Assumptions

To achieve the objective behind developing the proposed routing protocols, some assumptions need to be agreed to fulfil the requirements with which we are concerned in this thesis: these assumptions are as follows:

1. Each node in the network is equipped with a set of devices, which are considered to be available on vehicles at the present time. These include the On Board Unit (OBU), a preloaded digital road maps Global Position system(GPS) and Navigation System (NS). This enables each node in the network to provide surrounding neighbour nodes with information regarding changing circumstances during the journey.
2. Each node in the network is intending to utilise the knowledge of the SHD, by exchanging it with other nodes.
3. All nodes are willing to broadcast a HELLO beacon message based on cer-

tain circumstances (as in adaptive HELLO message), or periodically every 0.5 seconds (as with the routing protocols VSHDRP1 and VSHDRP2).

4. Each node obtains sufficient information about its position, direction, SHD and speed by exploiting the devices with which it is equipped, the standard format of the HELLO beacon message [54] is shown in table 3.1, while the format of the HELLO beacon message that used in VSHDRP is shown in table 5.1.
5. Each node in the network is willing to predetermine its route from the beginning of the journey by utilising the facilities incorporated in the NS.

Field Name	Description
NT	Node type
Node-seq	Node Identification of distinguished HELLO messages
Node-dir	Node First Direction
Node-pos	Node Current Position
Node-vlo	Node Moving Velocity

Table 3.1: the standard HELLO beacon message format in VANET

3.3.3.2 Network Models

The ad hoc network system that utilises the multi-hop techniques is initiated based on a set of nodes N , which move in the boundaries of a specific area A . In this type of network the scenario plays an important role in distributing the nodes inside the specified area A . It is possible to organise the node distribution in the ad hoc network based on certain types of applications, such as may be required for specialised use by the highways agency or for military movements. However in other applications the node movements can show a random distribution as in the rescue system.

In contrast to the two types of distribution for the nodes in ad hoc networks mentioned above, the distribution in VANET can fall between these two types (the organised and random distribution), the reason for this is that the nodes that are used in this type of network are vehicles, which can move randomly taking any direction; however these random movements are all necessarily restricted by the road network; therefore, the movement is predetermined by the road. Consequently the density of the nodes can show diversity depending on the application and in certain areas on the network because of traffic lights, an intersection or even an accident affecting the flow of vehicles on a road. The nodes density can be measured by equation 3.1.

$$\text{Density of nodes} = \frac{N}{A} \quad (3.1)$$

Suppose there are a number of nodes N moving in a certain area A , in some sections within the area the density of the node is high, at the same time in other sections the density of the nodes is very low, which sometimes leads to the generation of a disconnected area; in this case the network can be described using the terminology: Delay Tolerant Network (DTN); the disconnected area generated represents a barrier preventing delivery of the packet to the desired destination [12, 50, 16]. Due to the low number of nodes, it can then be difficult to find a node to act as an intermediate node to forward the packet to its destination.

Generally in an ad hoc network, each node in the network has a certain range of transmission called R , and each node uses the Omnidirectional antenna; the communication between any two nodes m and n (where $m, n \in N$) being considered a successful aspect of communication if each node is located in the

transmission region of the other, by satisfying the following two conditions:

- The condition $d_{mn} < R$, if we know that d_{mn} is representing the distance between m and n .
- If the receiving node n has an interference range R_{int} , any node ($z \in N$) that is with R_{int} of the receiving node n , $d_{zn} \leq R_{int}$ is not transmitting, generally the interference range R_{int} and the transmission range R are the same.

In the case of IEEE 802.11 MAC protocols [55], there is a need for node m , which acts as a sending node to be free of interference, as there is a need to receive the link layer acknowledgement from the n which acts as a receiving node. Therefore, in IEEE 802.11 MAC protocols, a node z , when falling in the interference range of the nodes m or n , must not be transmitted.

In general, the problem of hidden and uncovered terminals falls under the consideration of MAC protocols. The presence of this problem is solely dependent on use of the wireless network rather than on the wired network. Owing to the simultaneous transmission of the hidden nodes, the problem of hidden terminal problems can occur, those nodes are not within the direct transmission range of the sender, but neither are they hidden from the receiver (within the transmission range of the receiver). Therefore if two nodes have transmitted packets simultaneously, without any knowledge of the other's transmission, a packet collision at the receiving side will occur.

As can be seen in Figure 3.1, both node N_i and node N_j tried to transmit a packet to node N_x at the same time. This is because neither of them was

aware of the transmission of the other node as both nodes were hidden from each other. This means that both node packets collided at node N_x .

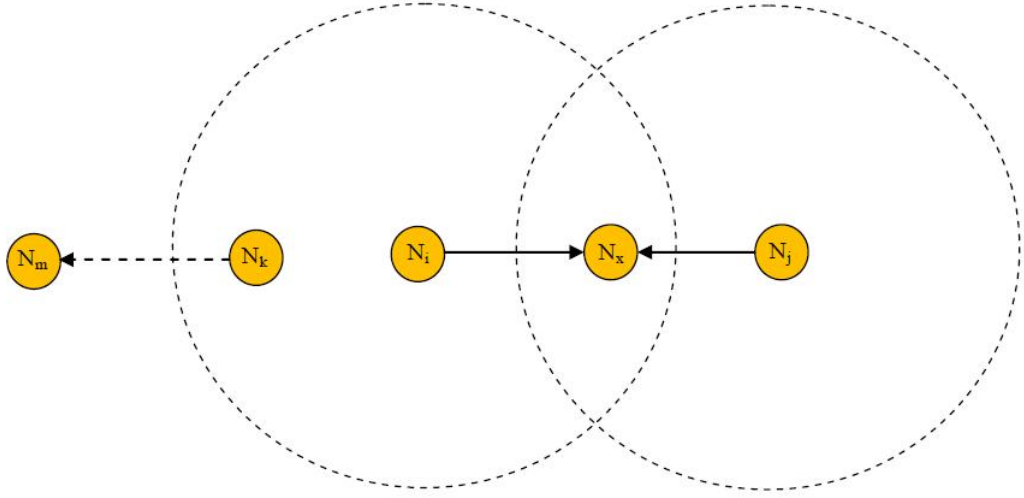


Figure 3.1: The collision of packets in simultaneous transmission

The exposed terminal problem relates to the phenomenon of one node blocking another node from transmitting due to its proximity to a transmitting node. Figure 3-1 illustrates that the node N_k cannot transmit to node N_m because the transmission from node N_i to node N_x is already in progress and the transmission of N_k would interfere with the on-going transmission from node N_i .

The performance of ad hoc networks is reduced by the occurrence of hidden and exposed terminal problems, especially when traffic load is high. Therefore, reducing the transmission of unnecessary and redundant packets will effectively limit this problem by reducing the number of HELLO beacon messages between the nodes, by implementing the techniques of adaptive HELLO beacon message in VANET especially in a high density environment.

3.3.3.3 Mobility Features

The mobility feature is considered to be one of the vital characteristics making VANET a unique subgroup in ad hoc network applications [18]. Due to the high mobility in VANET, the topology tends to change frequently [56]. Accordingly the proposed routing protocols, VSHDRP that have been introduced in this thesis, involve taking this characteristic into consideration, through exploiting knowledge about the nodes in the network, allowing them to be utilised for the filtration process that has been designed for the VSHDRP mechanism.

In VANET, The mobility model is a unique case as compared with MANET; this is because the nodes are vehicles, and those vehicles need to travel on a selection of limited roads. In some ways the nodes movements are therefore considered to be organised and predictable, in contrast to node movements in MANET, which can be expressed as a random and unpredictable.

3.3.3.4 Traffic Model

The successful transmission process from the sender node s to the receiver node r is significantly dependent on the Signal-to-Noise Ratio (SNR), if the SNR at the receiver node, is at the boundaries of the threshold's limitations ($SNR_{sr} \geq SNR_{threshold}$). Mainly the tracking discussed in this thesis relates to the network layer, for that reason the physical layer is not our concern, as that remains an area to consider in future work. The type of antenna used by each node in the network is the Omnidirectional antenna.

3.4 Overview of context aware system

The context-aware system provides users with new facilities to sense tangible and intangible activities and afford realistic reasoning according to current conditions, in order to achieve unrealised functionalities and assist in accomplishing tasks precisely without providing more concerns from the user's side.

The definition of context can be expressed thus: if a certain situation relating to any entity, such as a person, place or an object, is considered to be related to an application interactively it is then affected by any available information [57]. The context has been modelled based on six key determinants depending upon the environmental types; physical environment (what, where and when), ICT (Information and Communication Technology) environment (how) and User environment (who and why) [58] [59, 60] . The definition of each of these determinants is based on the environment as described in details as follows:

- Context of physical environment: this comprises three determinants as follows:
 - What: describing the type of physical environment the context aware system is operating in; concerning the light, temperature and chemical concentrations.
 - Where: describing the awareness based on the context location; such as having information about the destination location when on a pre-determined route.
 - When: specifying when the system needs to perform its context awareness; at what moment, after a while or according to a specific action or event.

- Context of ICT environment system: this type of environment involves one determinant:
 - How: this determinant relates to how the context is formed and characterised in the ICT environment, for example, when interacting with mobile devices through a wireless network.
- Context of user environment: the environment in this instance is related to the user, and operates according to two determinants.
 - Who: this determinant refers to descriptions of the circumstances of a person according to how they perform a certain activity.
 - Why: indicates the intention behind making the context useful.

A context aware system can be defined as a system that can take a decision about doing a certain action after undertaking a series of processing by sensing circumstances through, for instance, a fitted set of sensors automatically detecting the distance of the camera to the specific object, so as to adapt the focus [60].

3.5 The Stages of Context-aware System

In this section the main components of the context aware system, which is considered as the basis for any system are described in detail. In general, a context aware system comprises three main connected stages of functionalities: sensing, processing and acting as shown in figure 3.2. Systems have different levels of sophistication when presenting these functionalities; some of them can investigate a wide range of sensors but provide less concentrated reasoning. However, other system can employ minimal while undertaking a lot of processing activity before acting [61]. The three main phases of a context

aware system are discussed as follows:

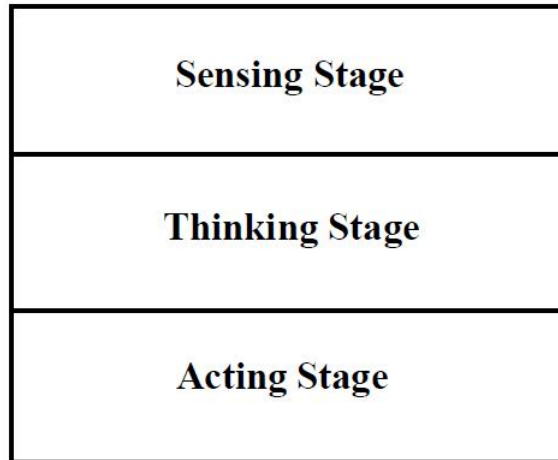


Figure 3.2: Stages of context aware system

3.5.1 Sensing Stage

Any type of sensor can be utilised to aggregate required information that is relevant to a notion of the physical world. The knowledge acquired can be exploited by a computer system to act appropriately in relation to a specific situation. A more general view of the surrounding conditions in the physical world can be observed by utilising a combination of sensor sets, which can provides obvious clear and precise outlook.

On the one hand many types of sensors have emerged to provide such type of knowledge, including light sensors, movement sensors and temperature sensors. On the other hand there are certain types of devices that may act as sensors although not designed for such a purpose; for example the timer in a computer or a microphone. Vehicle manufacturers are one of the most important sectors in terms of consumption of a wide variety of sensor types, in order to provide users with a safe and comfortable driving experience through-

out their journey, by making increasing the number of functions that can be performed automatically; such as darkness detection to turn headlights on, switching wipers on automatically and alerting drivers to possible hazards on the road ahead.

In general the sensory information obtained at this stage can result not only from a hardware device; indeed, sensors can be categorised according to three different groups [62]:

- Physical sensors: Those sensors used to collect physical data.
- Virtual sensors: using services and application from different useful software, such as detecting a person's location based on their use of email or browsing activity.
- Logical sensors: in this type physical and virtual sensors are employed together as sources of information, to assist in finding a user's current position by exploiting the login PC in combination with the database mapping information from devices.

3.5.2 Thinking Stage

Once the sensors have performed their task and collected the required information pertaining to a certain situation, this information, which may be available in different forms, is then employed at this stage to be utilised in a series of processing and reasoning activities to take a decision about the most appropriate action. The processing is required to make the information collected more obvious after it has been aggregated. Simultaneously the storage and management task is being performed at this stage; this is responsible for managing the data to make it available through an active interface [61, 62].

3.5.3 Acting Stage

At this stage, once the situation has been recognised and the decision has been taken to act on the basis of processing and dealing with the knowledge gathered in the first stage, then this action takes place to fulfil and complete the execution of that decision.

3.6 VSHDRP Routing Protocol Evaluation

The VSHDRP is fully formalised in the Calculus of Context-aware Ambients (CCA), which is a process using calculus for modelling and analysing mobile and context-aware systems, for the sake of evaluating the new protocol behaviour and testing the validation of the protocol algorithm. The behaviour of the protocol is then analysed via a number of scenarios using the CCA interpreter. Such an analysis is useful for rapid prototyping and validation of the key properties of the protocol 3.3.

3.6.1 CCA Based Evaluation

The VSHDRP has been formalised by using the CCA, in order for evaluating its behaviour and testing the validation of the algorithm.

3.6.1.1 The Calculus of Context-aware Ambients(CCA)

CCA is a process calculus for modelling context-aware systems. The main features of the calculus include *mobility*, *context-awareness* and *concurrency*. The concept of *ambient* is an abstraction of a place where computation may happen. An ambient can be mobile and can contain other ambients called

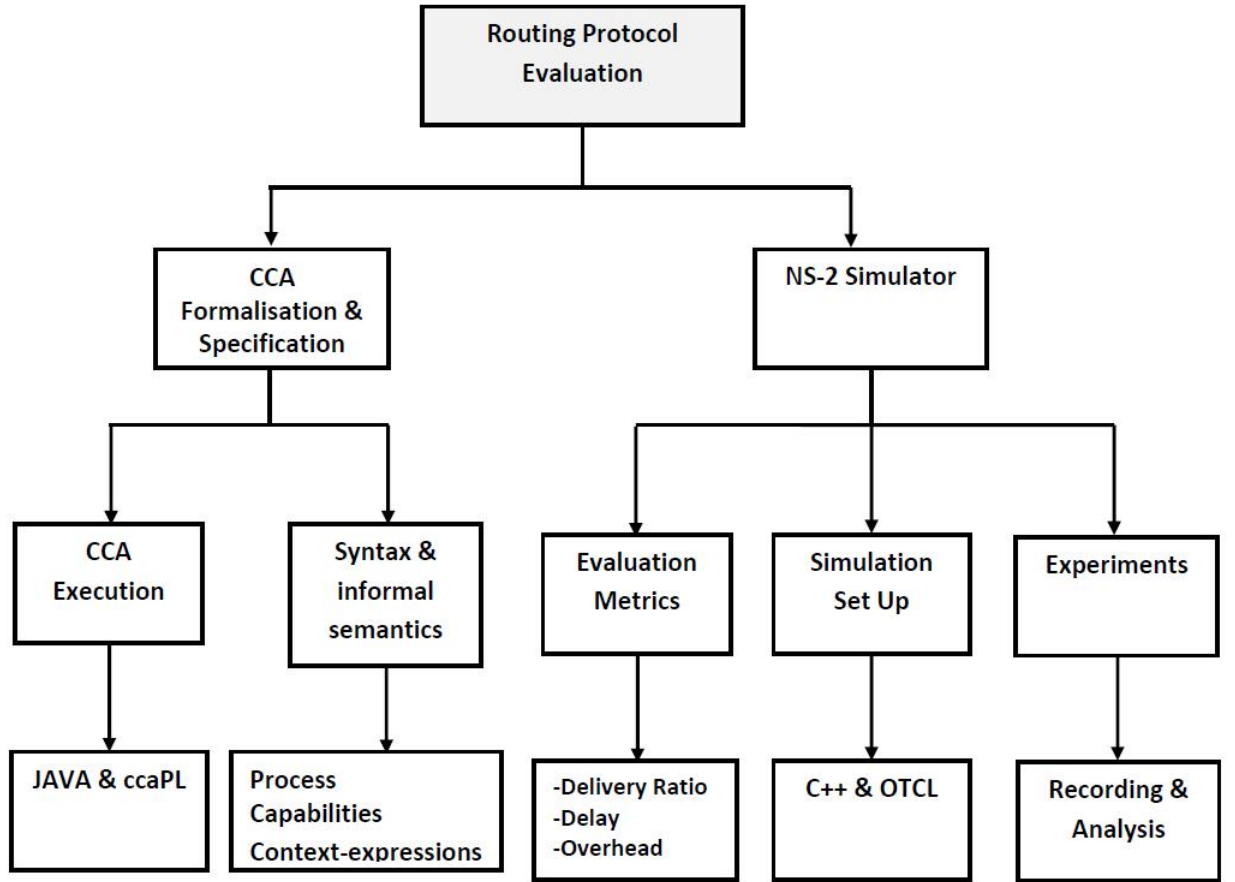


Figure 3.3: Routing mechanism evaluation

child ambients organised in a tree structure. Such a hierarchy can be used to model any entity in a pervasive system –whether physical, logical, mobile or immobile– as well as the environment (or context) of that entity. In addition to child ambients, an ambient can also contain a process specifying the capabilities of that ambient, i.e. the actions the ambient is allowed to perform, such as mobility capabilities, context-aware capabilities and communication capabilities. Moreover, specifications written in **CCA** are executable. All these features have motivated our choice of **CCA** for the specification and analysis of the Vehicular Second Heading Direction Routing Protocol (VSHDRP).

This section presents the syntax and the informal semantics of **CCA**. Due to

the space limit, only features relevant to our work are presented. We refer interested readers to [8] for the full details of the calculus. Table 3.2 depicts the syntax of **CCA**, based on three syntactic categories: processes (denoted by P or Q), capabilities (denoted by M) and context-expressions (denoted by E). Like in the π -calculus [63, 64], the simplest entities of the calculus are *names*. We assume a countably-infinite set of names, elements of which are written in lower-case letters, e.g. n , x and y . We let \tilde{y} denote a list of names and $|\tilde{y}|$ the arity of such a list. We sometimes use \tilde{y} as a set of names where it is appropriate.

P, Q	$::=$	$\mathbf{0} \mid P Q \mid (\nu n) P \mid !P \mid n[P] \mid \{P\} \mid E?M.P$ $\mid \mathbf{find} \tilde{x} : E \text{ for } P$
M	$::=$	$\mathbf{in} n \mid \mathbf{out} \mid \mid \alpha \mathbf{recv}(\tilde{y}) \mid \alpha \mathbf{send}(\tilde{y}) \mid \mathbf{del} n$
α	$::=$	$\uparrow \mid n\uparrow \mid \downarrow \mid n\downarrow \mid :: \mid n :: \mid \epsilon$
E	$::=$	$\mathbf{True} \mid \bullet \mid n = m \mid \neg E \mid E_1 E_2 \mid E_1 \wedge E_2 \mid$ $\oplus E \mid \diamond E$

 Table 3.2: Syntax of **CCA** [8]

3.6.1.1.1 Processes The process $\mathbf{0}$, aka *inactivity process*, does nothing and terminates immediately. The process $P|Q$ denotes the process P and the process Q running in parallel. The process $\nu n P$ states that the scope of the name n is limited to the process P . The replication $!P$ denotes a process which can always create a new copy of P , i.e. $!P$ is equivalent to $P|!P$. Replication can be used to implement both iteration and recursion. The process $n[P]$ denotes an ambient named n whose behaviours are described by the process P . The pair of square brackets '[' and ']' outlines the boundary of that ambient. The process $\{P\}$ behaves exactly like the process P .

A context expression specifies the condition that must be met by the environment of the executing process. A *context-guarded prefix* $E?M.P$ is a process that waits until the environment satisfies the context expression E , then performs the capability M and continues like the process P . The dot symbol ‘.’ denotes the sequential composition of processes.

We let $M.P$ denote the process $\mathbf{True}?M.P$, where \mathbf{True} is a context expression satisfied by all context. A *search prefix* $\mathbf{find} \tilde{x} : E \text{ for } P$ is a process that looks for a set of names \tilde{n} such that the context expression $E\{\tilde{x} \leftarrow \tilde{n}\}$ holds and continues like the process $P\{\tilde{x} \leftarrow \tilde{n}\}$, where the notation $\{\tilde{x} \leftarrow \tilde{n}\}$ means the substitution of n_i for each free occurrence of x_i , $0 \leq i < |\tilde{x}| = |\tilde{n}|$.

3.6.1.1.2 Capabilities Ambients exchange messages using the output capability $\alpha \mathbf{send}(\tilde{z})$ to send a list of names \tilde{z} to a location α , and the input capability $\alpha \mathbf{recv}(\tilde{y})$ to receive a list of names from a location α . The location α can be ‘ \uparrow ’ for any parent, ‘ $n \uparrow$ ’ for a specific parent n , ‘ \downarrow ’ for any child, ‘ $n \downarrow$ ’ for a specific child n , ‘ $::$ ’ for any sibling, ‘ $n ::$ ’ for a specific sibling n , or ϵ (empty string) for the executing ambient itself. The mobility capabilities **in** and **out** are defined as follows. An ambient that performs the capability **in** n moves into the sibling ambient n . The capability **out** moves the ambient that performs it out of that ambient’s parent. The capability **del** n deletes an ambient of the form $n[0]$ situated at the same level as that capability, i.e. the process $\mathbf{del} n.P \mid n[0]$ reduces to P . The capability **del** acts as a garbage collector that deletes ambients which have completed their computations.

Example 3.6.1

- The process $n[\mathbf{in} m.\mathbf{out}.0] \mid m[\mathbf{in} n.\mathbf{out}.0]$ describes the behaviours of two sibling ambients n and m concurrently willing to move in and out of

one another.

- The ambient $n[(\blacklozenge \mathbf{at}(m))? :: \mathbf{send}(msg).\mathbf{0}]$ releases the message ‘ msg ’ only when at location m ; where the context expression $\blacklozenge \mathbf{at}(m)$ holds if n is a child ambient of the ambient m . The formal definition of the predicate \mathbf{at} is given in Example 3.6.2.

3.6.1.1.3 Context model In CCA, a context is modelled as a process with a hole in it. The hole (denoted by \odot) in a context represents the position of the process that context is the context of. For example, suppose a system is modelled by the process $P \mid n[Q \mid m[R \mid S]]$. So, the context of the process R in that system is $P \mid n[Q \mid m[\odot \mid S]]$, and that of ambient named m is $P \mid n[Q \mid \odot]$. Thus the contexts of CCA processes are described by the grammar in Table 3.3. Properties of contexts are called context expressions (CEs in short).

$$E ::= \mathbf{0} \mid \odot \mid n[E] \mid E \mid P \mid (\nu n) E$$

Table 3.3: Syntax of contexts

3.6.1.1.4 Context expressions The CE **True** always holds. A CE $n = m$ holds if the names n and m are lexically identical. The CE \bullet holds solely for the hole context, i.e. the position of the process evaluating that context expression. Propositional operators such as negation (\neg) and conjunction (\wedge) expand their usual semantics to context expressions.

A CE $E_1 \mid E_2$ holds for a context if that context is a parallel composition of two contexts such that E_1 holds for one and E_2 holds for the other. A CE $n[E]$ holds for a context if that context is an ambient named n such that E

holds inside that ambient. A CE $\oplus E$ holds for a context if that context has a child context for which E holds. A CE $\diamond E$ holds for a context if there exists somewhere in that context a sub-context for which E holds. The operator \diamond is called *somewhere modality* while \oplus is aka *spatial next modality*.

Example 3.6.2 *We now give some examples of predicates that can be used to specify common context properties such as the location of the user, with whom the user is and what resources are nearby. In these sample predicates we take the view that a process is evaluated by the immediate ambient λ say that contains it.*

- $\text{has}(n) \hat{=} \oplus (\bullet \mid n[\mathbf{True}] \mid \mathbf{True})$: holds if ambient λ contains an ambient named n ¹
- $\text{at}(n) \hat{=} n[\oplus(\bullet \mid \mathbf{True})] \mid \mathbf{True}$: holds if ambient λ is located at an ambient named n
- $\text{with}(n) \hat{=} n[\mathbf{True}] \mid \oplus(\bullet \mid \mathbf{True})$: holds if ambient λ is (co-located) with an ambient named n

3.6.2 NS-2 Based Evaluation

There is a necessity to test the performance of a VSHDRP mechanism, measure the ratio of delivered packet, and then calculate the generated overhead traffic obtained from applying the SHD aspect in VSHDRP mechanism and the average time delay needed to deliver the packet to its designation. For that reason, it is important to elect an appropriate network simulator, in order to examine the performance mechanism of the proposed routing protocol.

A validation for applying the NS-2 simulator on the VSHDRP mechanism will

¹The symbol ' $\hat{=}$ ' means *defined by*.

be introduced, and provide the way of determining the simulated environment and metrics explaining the parameters already set, as shown in figure 3.3.

3.6.2.1 Network Simulator-2 (NS-2)

The performance of the VSHDRP needs to be tested, to show its effectiveness in relation to other proposed routing protocols in VANET, therefore one of the simulation tools needs to be used for this purpose. The work introduced in this thesis has been tested using the Network Simulator-2 (NS-2).

Various simulation tools have been utilised to evaluate and simulate the performance of routing protocols in VANET. Among those tools are for example the Network Simulator NS-2 [65], global Mobile System Information System Simulator Library GloMoSim [66] and OPNET Modeller [67], either C++ or the Java programming language are used to build the simulators; some other approaches have also been tested using self-developed codes. Kurkowski et al. [1] found that the NS-2 is the network simulator used in the majority of ad hoc network research, although other simulation tools are available. 43.8 percent have used the NS-2 simulator as a simulation tool to examine their approaches, as shown in the figure 3.4.

In this section, VSHDRP is simulated using the Network Simulator NS-2 for the purpose of evaluating and comparing it to other routing protocols. The reason behind selecting the NS-2 as the simulator for this work, is that it provides a range of characteristics that make it a distinguished simulator tool, NS-2 is an open source code simulator that can be easily modified and expanded [65].

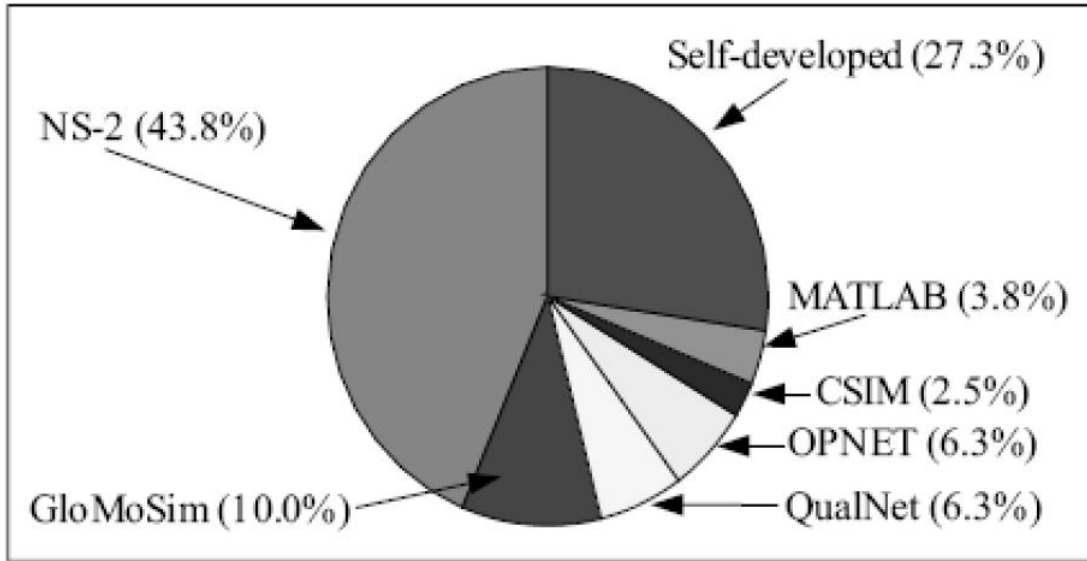


Figure 3.4: Simulator usage from MobiHoc survey [1]

The University of California at Berkeley and the VINT project [68] have developed the NS-2, which is a discrete event and an object-oriented simulator designed for networking research. Several different NS-2 versions have been released over the last few years; the latest version is the NS-2.34. The NS-2 provides significant support for the simulation of TCP, routing, and multicast protocols over wired and wireless networks.

By using OTcl scripts facility programming, a specific network topology can be declared, in addition the application and protocols that need to be simulated can be achieved, as can a determination of how to define the final format of the output obtained.

Through the application of an OTcl linkage the objects compiled in C++ can be used. For that reason from the user's perspective, the NS-2, can be inputted using the OTcl scripts as input commands into the interpreter; obtaining output in the form of trace files as illustrated in figure 3.5.

The NS-2 simulator is written in C++, with an OTcl (Object Tool Command

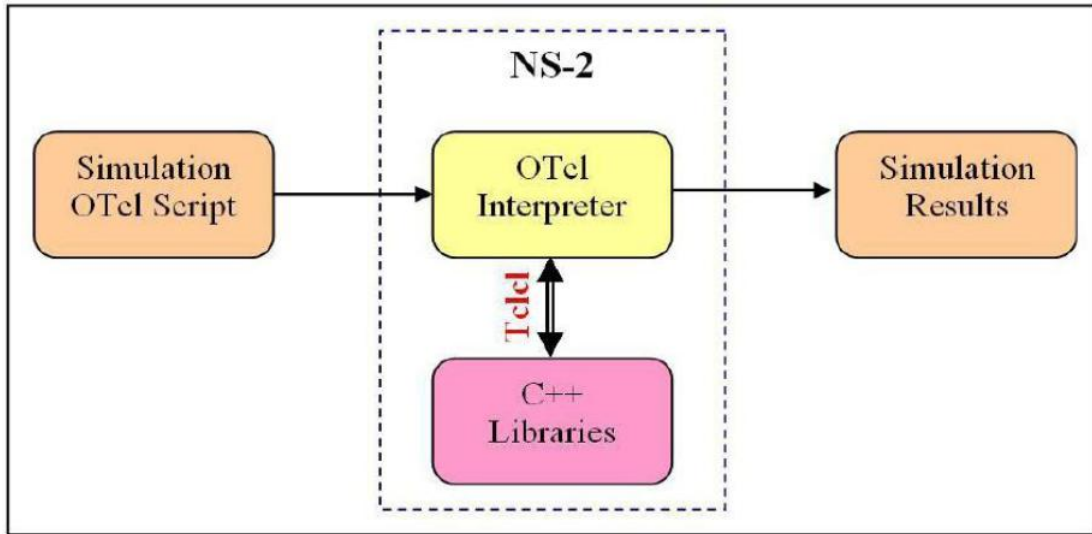


Figure 3.5: NS-2 schematics structure [2]

Language) interpreter as a command and configuration interface. C++ is fast to run but slower to change, and these qualities make it appropriate for use for comprehensive protocol implementation. In contrast, the OTcl part, which runs much slower, but can be changed very quickly, is used for the simulation configuration. The split-language programming approach allows for fast generation of large scenarios, which is considered to be one of NS-2 simulator advantages [2].

3.7 Simulation Environment of the Routing Protocol Mechanism

A modification has been applied to the TCL script file for mobile ad hoc wireless simulations afforded by the NS-2 distribution, in order to fit the simulation environment of the routing protocol; each node in the network includes a set of network components, which must be declared. These components, contained by each network node, are; MAC layer, Link Layer (LL), Interface

Queue (IfQ) and the wireless channel responsible for transmitting and receiving signals. Furthermore, it is important to determine other parameters such as antenna type, the model of the radio-propagation and the type of routing protocol; in our case we use the VSHDRP as the type of routing protocol, in order to test its performance on the network.

The proposed routing protocol needs to be evaluated by the NS-2 simulator based on specified metrics; those metrics can be as follows. The *efficiency ratio of packet delivery* computes the ratio of delivered packet to the destination. *End-to-End time delay* computes the average time delay required when delivering the packet to its destination. *Overhead* computes the accumulated number of packets transmitted over the network. Three different scenarios have been applied to simulate the proposed protocol, using the above-mentioned metrics; the network size scenario using a variety number of nodes (10, 20, 50,100,150), the data packet size, with a different packet capacity (10 b, 500 b, 1000 b, 2000 b, 4000 b), the HELLO beacon message interval (0.2 sec, 0.5 sec, 1.0 sec and 5.0 sec), and to be fair enough in comparing the VSHDRP result with GPSR result, the traffic pattern used in the simulation is the constant bit rate (CBR) as the GPSR used this type of traffic pattern in simulation. The performance of the routing mechanism can be affected by other factors during the simulation time, which will be illustrated in details in the following chapters. The parameters for prescribing the value of implementing the VSHDRP are shown in table 3.4.

Parameters	Network Size Scenario	Data Packet Size Scenario	Vehicle beacon interval Scenario
No. of Nodes	10, 20, 50, 100, 150	10	10
Max. Vehicle Speed	25-75 m	25-75 m	25-75 m
Simulation Time	250 sec	250 sec	250 sec
Network Space	4000X3200 meter	4000X3200 meter	4000X3200 meter
Radio range	250 meter	250 meter	250 meter
MAC Protocol	IEEE 802.11	IEEE 802.11	IEEE 802.11
Radio Propagation Model	Free Space/ Two Ray	Free Space/ Two Ray	Free Space/ Two Ray
Data Packet Size	10b	10 b - 4 Kb	10b
Antenna Model	Omni Antenna	Omni Antenna	Omni Antenna
Traffic Pattern	CBR	CBR	CBR
Vehicle Beacon Interval	5.0 sec	5.0 sec	0.2 sec, 0.5 sec, 1.0 sec, 5.0 sec

Table 3.4: NS-2 simulation Setup

3.8 Summary

In this chapter, a routing management solution in VANET has been introduced; its purpose being to mitigate the obstructions and provide improvements over other management solution. Circumstances and environmental situations are required to carry out the algorithm of routing protocol have been described.

Therefore an overview of new aspects of routing managements in VANET has been introduced in this chapter. The mechanism of this proposed routing protocol has been developed in order to achieve the objective of this thesis, which can be represented by improving the routing managements in VANET. The proposed solution has been introduced in two stages; the first stage, is the adaptive HELLO beacon message based on the aspect of context awareness system, while the second stage provides a novel routing protocol in VANET (VSHDRP). In addition an overview of the evaluation tools has been introduced which can be presented by the CCA and NS-2.

Moreover, a synopsis of the context-aware system has been given to provide sufficient understanding to support applying the aspects of this system to the routing management; particularly in regards to the adaptive HELLO beacon message, which is based on the three main phases of context-aware system, representing; sensing, processing and reacting.

Chapter 4

On Board Unit Architecture Based on Context-aware System

Objectives:

- Determining the objective behind developing the adaptive HELLO beacon message technique
 - Defining the components of OBU architecture based on the context-aware system
 - Describing the technique of the adaptive HELLO beacon message
 - Describing the three phases of the architecture that have been designed based on the concept of context-aware system
-

4.1 Introduction and Motivation

In this chapter a novel architecture for OBU in VANET is introduced based on the concept of the context-aware system [69]. It is built based on a new technique for sending the HELLO beacon message adaptively in a VANET environment. This architecture shows the three main stages undertaken by the context-aware system represented by; sensing, processing and reacting, in order to send the HELLO beacon message adaptively based on a specific vehicle's circumstances rather than on broadcasting the messages periodically. This is done so as to organise the broadcasting process of the HELLO beacon message, to avoid sending a redundant HELLO message, which will consequently improve the system's performance.

As mentioned previously in chapter three, the HELLO beacon message is considered as a vital technique in VANET [13], because it allows each node in the network to inform the neighbouring node about its situation, by providing them with a sufficient knowledge about its circumstances, including position, direction, speed and destination. Two main types of HELLO beacon messages can be raised; the first is to send the HELLO beacon message when there is a need to send a packet, this type falls under the type of reactive routing protocols (on demand), this suffers from long time processing especially with a high mobility network like VANET. The second is to send a HELLO beacon message periodically, for instance every 0.5 seconds, this type is considered to be more suitable for VANET [6].

However bandwidth consumption is the major problem with this type espe-

cially in the case of a high density network. Due to the limited range of bandwidth frequency (10-20 MHz) for VANET applications [70], the fair use of bandwidth has an impact on reducing the time delay for disseminating messages. If a vehicle needs to send a message and it is discovered that there are no opportunities for transmission, it must wait until there is an opportunity for a new transmission [33]; that will lead to an increase in the latency, especially in urban areas and with continuous increasing use of the application that is used in VANET, which requires sufficient bandwidth to be applied. For this reason sending a HELLO beacon message periodically has an impact on increasing the overhead, which then leads to waste of the limited bandwidth [13].

4.2 Overview of the adaptive HELLO Message

To tackle the limitation of sending HELLO messages periodically, this research proposes a novel technique as an alternative approach to organise the process of exchanging the HELLO beacon message between vehicles alongside the periodic approach and the on demand approach. The adaptive HELLO beacon message technique is based on governing the process of sending a HELLO beacon message according to the changing circumstances of each vehicle on the network. These circumstances could relate to the vehicle's direction or velocity. Owing to the restricted mobility in VANET that is restrained by roads, vehicles will keep moving in a specific direction until the end of the road, after which their direction will change and the distance between vehicles will remain fixed especially on highways.

As mentioned previously, the algorithm is based on two main factors; road di-

rection and the vehicle's velocity, if the value of either of these factors changes, the vehicle will broadcast a HELLO message; otherwise it will stop broadcasting the HELLO message and wait for another change.

4.2.1 Message Broadcast Conditions

This section describes when the HELLO beacon message needs to be sent, in order to notify surrounding vehicles about its situation and provide them with up to date information in case one of those vehicles intends to forward a packet. The situation involving sending HELLO beacon messages are set as follows:

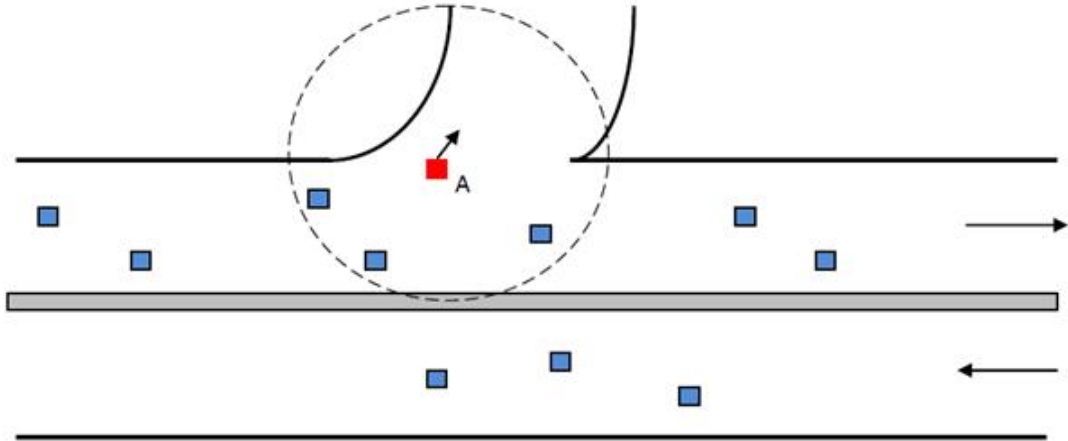


Figure 4.1: Broadcasting when the vehicle leaves the road

- If any of these vehicles changes its direction, as shown in figure 4.1 for instance the vehicle A takes the next exit; it will make a final broadcast on this road before leaving it, to notify other vehicles of its absence to avoid the unnecessary sending of any packets to this vehicle in future.
- If any of these vehicles changes its speed according to the predefined limit $S_{up} > S > S_{down}$, Where S , S_{up} and S_{down} are the current Vehicle's

Speed, maximum limit of speed and minimum limit of speed respectively.

For example if a breakdown is occurred, then the vehicle' speed will be reduced gradually until reaching the speed zero, that means the vehicle has exceed the minimum limit.

- If a new vehicle merges onto the road it will broadcast a HELLO beacon message to notify other vehicles regarding its existence on the road, then in turn the nearest vehicle will send a HELLO beacon message that includes information about itself and other vehicles on the road, this situation is shown in figure 4.2.

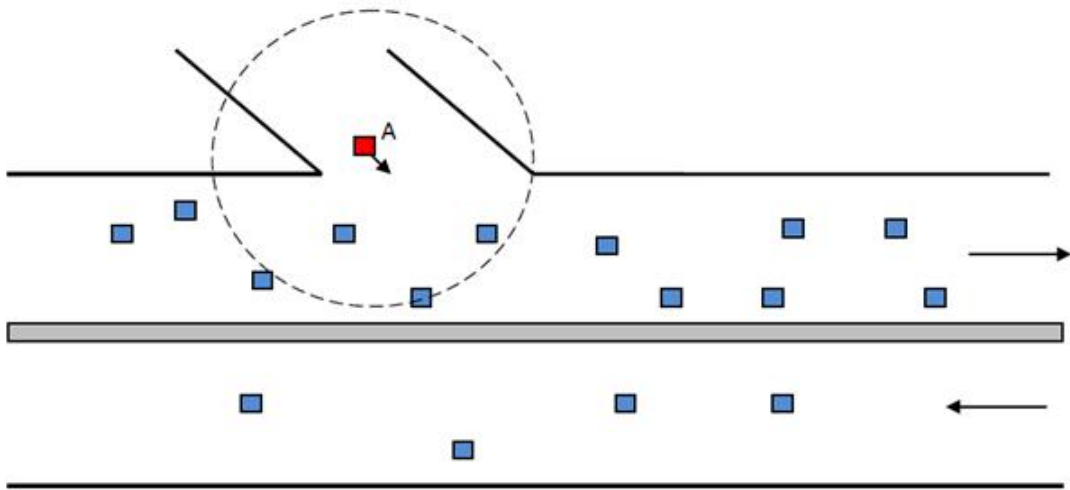


Figure 4.2: Broadcasting when the vehicle merges in the road

- If any vehicle referred to as A receives a HELLO beacon message about the existence of a new vehicle N in the road, this will lead to a change in the neighbours table of A , and then A in its turn will inform other vehicles (vehicles are not neighbour to N) about this change.

In this way all the vehicles in the road will have sufficient information about any other vehicle in the road; these conditions allow vehicles to predict a more precise route to their packets, which in turn has an impact on improving the

performance of the system:

4.2.2 Adaptive HELLO Beacon Message Scenario

In this section a simple example has been provided to explain how the conditions relevant to sending HELLO messages are occurring. Let us assume this scenario, if any of vehicles are travelling on a dual carriageway at the same time, each vehicle maintains its speed in a fixed range $S_{up} > S > S_{down}$ when travelling on the road, if any vehicle's speed exceeds these limits, then the HELLO beacon message will be sent. According to that situation vehicles that belong to this group do not need to send HELLO beacon messages in a periodic way to notify other vehicles about their existence, because there is no change in the network topology.

In addition, changing the direction of the vehicle by changing the road has an impact on the network topology, because that will make this vehicle out of the communication ranges of other vehicles in the road. Thus a HELLO beacon message will be sent by this vehicle to notify other surrounding vehicles about its new situation.

4.2.3 Adaptive HELLO Message Mechanism

In this section the mechanism of the adaptive HELLO message is described, in order to show the effectiveness of this process as influenced by vehicle movements depending upon specific parameters. Flowchart in figure 4.3 shows the process of sending a HELLO message adaptively. Here, on the one hand the vehicle will sense the change in its direction from the GPS system that is connected to the OBU through the Direction Control and Monitor Unit (DCMU),

which is responsible for collecting information about direction from the GPS and comparing it with predefined digital road maps, which are stored in a database and storage unit; after processing this information through the core unit this represents the main OBU processor. The decision can be taken in order to decide if the next action (sending HELLO beacon message) needs to be taken.

On the other hand, brake pedal sensor, fuel pedal sensor and the other types of sensors are connected to the OBU through the Speed Control and Monitor Unit (SCMU), which is responsible for sensing and observing the changes in speed, according to the readings on the sensors, the SCMU compares this value with a threshold of speed limits $S_{up} > S > S_{down}$, if any exceeding of these limits occurs, the decision will be taken to send HELLO beacon message, to inform neighbouring vehicles about this change in vehicle circumstances, which may lead to a change in the network topology.

In these two cases the vehicle will automatically send a HELLO beacon message to neighbouring vehicles with up to date information. This technique suggests sending the HELLO beacon message when the vehicle circumstances change (e.g. velocity or direction), this process will be based on the context-aware system when sensing that the change in circumstances is happening instantly, it is a self organising process and is part of the In-Vehicle communications, which are one of the main three categories in VANET.

Moreover each vehicle can calculate the current positions of the other vehicles by exploiting the information about speed and the direction of each vehicle as

provided by the HELLO beacon message.

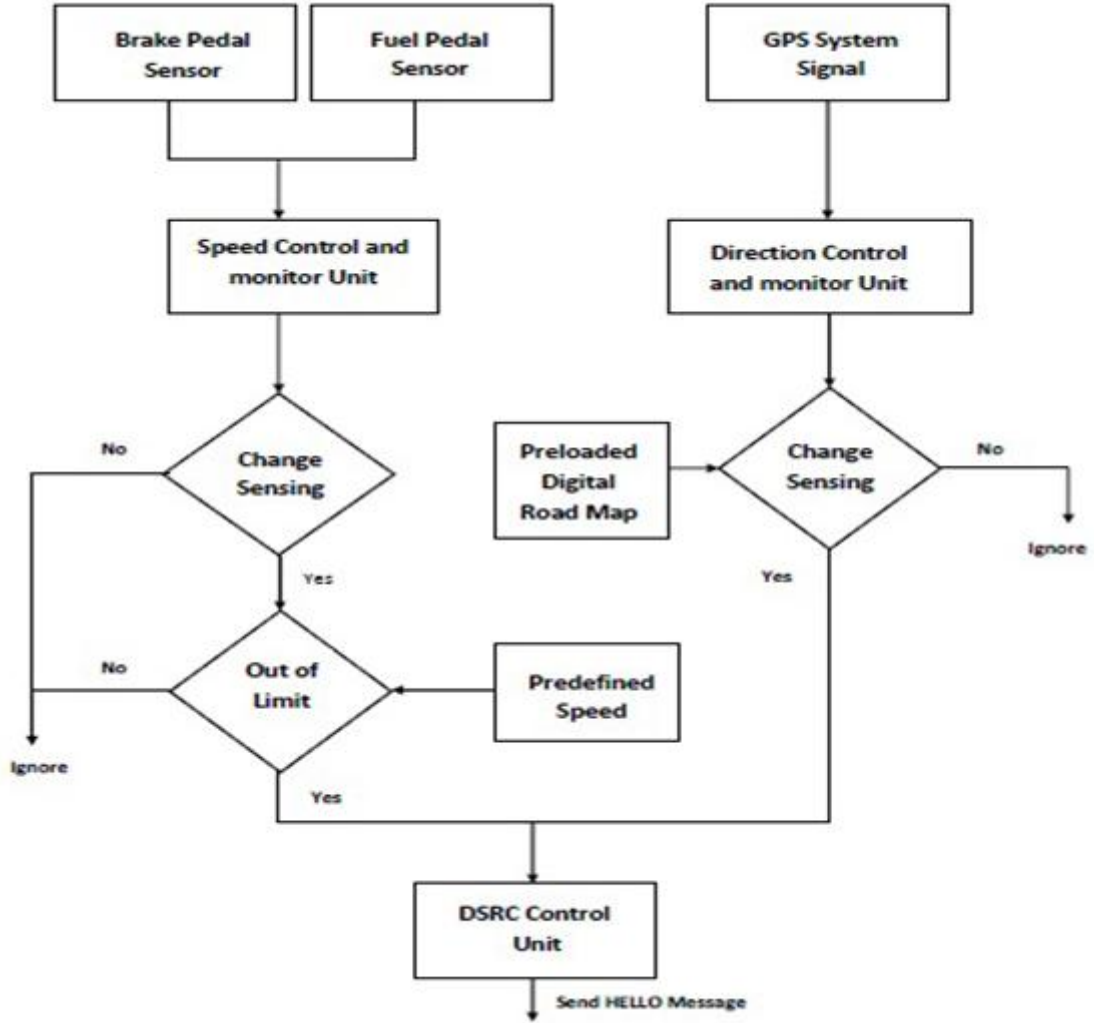


Figure 4.3: Sending HELLO message adaptively

4.3 OBU Architecture

This section describes the whole architecture of the adaptive HELLO beacon message [69] as illustrated in figure 4.4, it provides detailed definitions of each block in the architecture and reveals the functions performed by each unit in the architecture, in particular how these units interact with each other to accomplish the task of sending the HELLO beacon message. In the architecture

blocks that appear in green colour represent a new units, which added in this thesis to the originals components of the OBU.

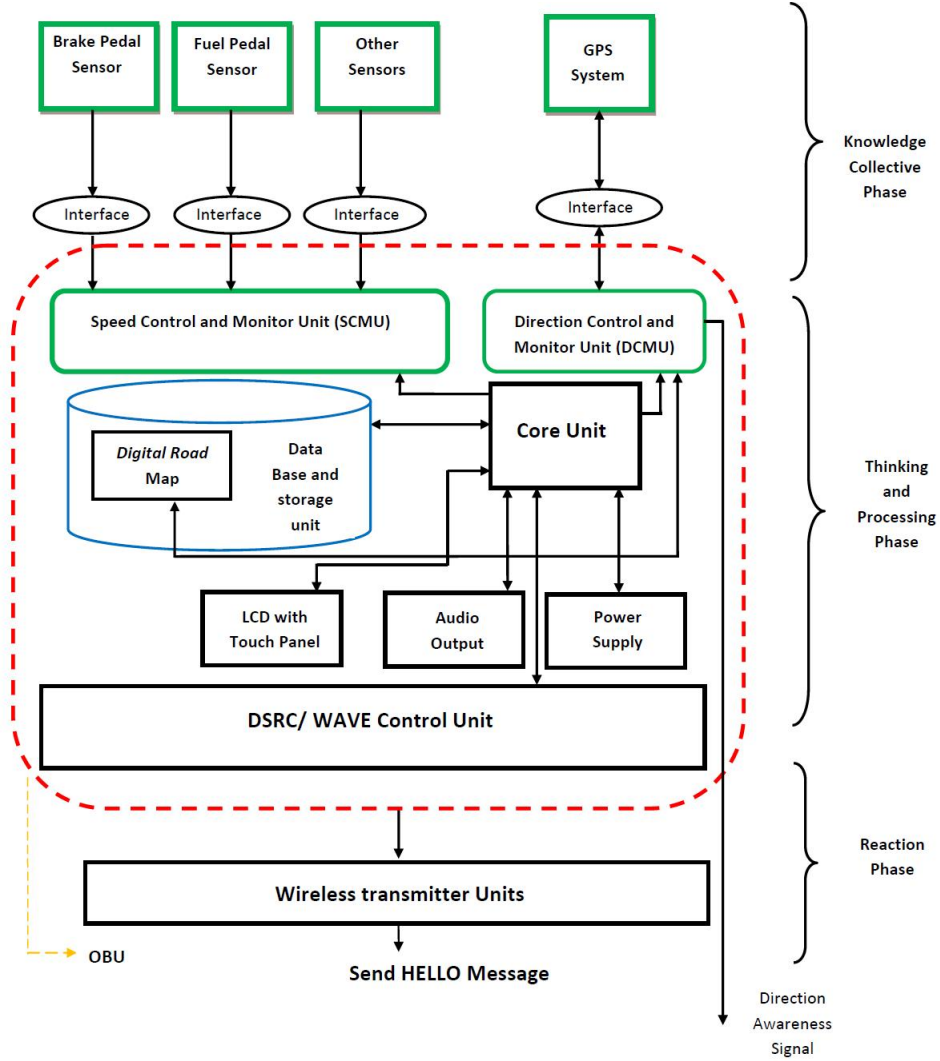


Figure 4.4: OBU architecture for adaptive HELLO message based on context-aware system

The mechanism for sending the HELLO beacon message adaptively in VANET can be represented according to the concept of the context-aware system [69], as shown in figure 4.5 which comprises the three main phases of the process; the collective knowledge phase, processing phase and reaction phase. Performing the task for the third phase depends on obtaining the output of the second phase, which in turn depends on obtaining the output from the first phase.

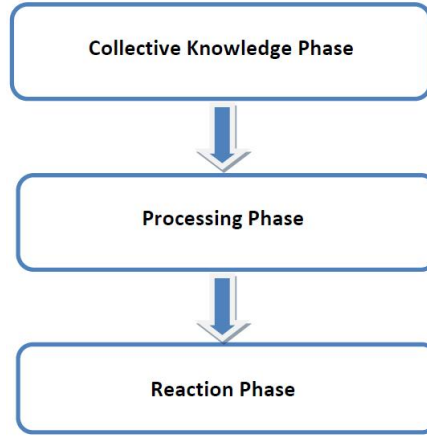


Figure 4.5: The three phases architecture

In other words the consequence of this process is based on the first stage for sensing the change through the fitted sensors in the vehicle, then in the second stage it process the values of these sensors to decide if these values indicate the possibility to affect the network, once the decision has been taken, the third stage plays the role of transmitting the HELLO beacon message.

4.3.1 Collective Knowledge Phase

This section shows how the knowledge can be sensed and what kind of application can be comprised. This phase is responsible for gathering the data required to be processed later in the next phase (processing phase), which represents the sensing subsystem in the context-aware system, as shown in figure 4.6.

This application exploits the knowledge acquired from the GPS system and a set of sensors fitted in vehicles, in order to collect any information that may affect the vehicle's circumstances such as direction and velocity. The main function of this type of application comprises two main types of knowledge as

shown in figure 4.4:

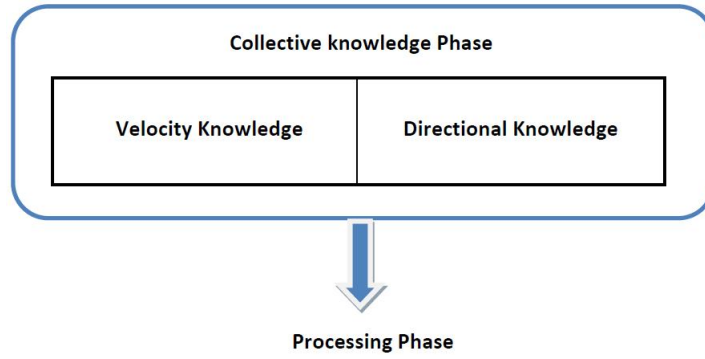


Figure 4.6: Collective knowledge phase

- **Direction Knowledge:** this type of knowledge can be acquired from the GPS system.
- **Velocity Knowledge:** this type of knowledge can be obtained by exploiting sensors fitted on a specific vehicle parts that can sense the change in vehicle speed. These can be divided via the sensors below:
 - *Brake Pedal Sensor:* a sensor fitted in the brake pedal; the main task of which is to sense any sudden brake and send a signal to the Speed Control and Monitor Unit (SCMU) in the OBU, which may cause a reduction in the vehicle's velocity immediately, as a result leading to a change in the network topology.
 - *Fuel Pedal Sensor:* a sensor attached to the fuel pedal is responsible for sensing any pressure applied on the pedal and sending the signal to the SCMU unit in the OBU. This signal is processed by the OBU to determine if the network topology will be affected.
 - *Other Sensors:* any sensors that may be attached to any part of the vehicle and have an impact on determining the occurrence of speed change; for instance the power cut sensor which is responsible for sensing the vehicle's power, and tyre pressure sensor which measures

the minimum limits of pressure required for the tyres to allow the vehicle to maintain its current speed.

4.3.2 Processing Phase

This phase represents the processing subsystem in the context-aware system. This phase acts as the key for our system (On Board Unit), as shown in figure 4.7; it processes the information collected in the previous stages (collective knowledge phase), to determine the action needed to be taken and in what part of the system. This phase involves the following units:

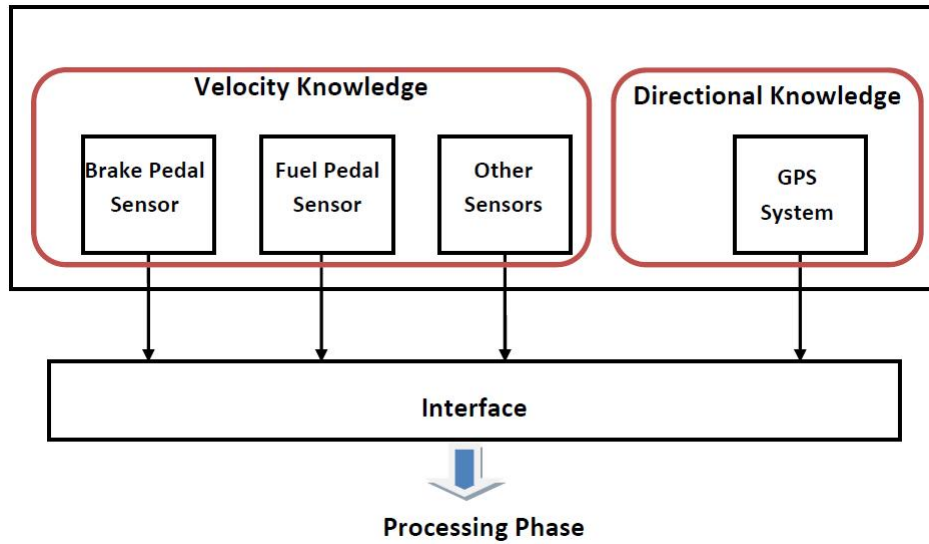


Figure 4.7: Active safety applications

- *Direction Control and Monitor Unit (DCMU)*: This element is responsible for dealing with the data collected about the vehicle direction in view of controlling the usage of this information by determining whether this information will lead to a change in the network topology. This task is mainly accomplished by connecting to a GPS system, in order to obtain required information about the vehicle direction.

- *Speed Control and Monitor Unit (SCMU)*: this unit is responsible for processing the data gathered in relation to the changing vehicle speed. Therefore this part is connected directly to a set of sensors; such as the brake pedal sensor and the fuel pedal sensor. By continuous observation of vehicle speed information, this unit can take the decision instantly to send a HELLO beacon message to neighbouring vehicles, when a significant change in vehicle speed has been detected according to predefined limits.
- *Data Base and Storage system*: this unit represents the main storage system in the OBU, and contains the preloaded digital road map unit that can be used with the information gathered from the GPS system to determine the direction of the vehicle and the complete route for the journey.
- *Core Unit*: this unit represents the processor of the OBU; controlling all tasks in the OBU and coordinating all activities between the units inside the OBU.
- *LCD with touch panel unit*: this is a Graphical User Interface (GUI); it represents the interface with the user to interact with the system.
- *Audio Output*: this aims to provide the driver with more concentration on the road; this type of output is considered to be useful.
- *Power Supply*: this is the main unit responsible for controlling power in the system.
- *DSRC/WAVE Control Unit*: Dedicated Short range Communications, which is responsible for organising the usage of the DSRC signals.

4.3.3 Reaction Phase

This phase represents the acting subsystem in the context-aware system. Once the information gathered is collected and the situation is recognised, the decision is taken to send the data (HELLO message) through to the transmitter units.

4.4 Summary

In this chapter, a novel OBU architecture based on the adaptive HELLO beacon message in VANET has been introduced, in order to reduce the number of HELLO beacon messages exchanged between vehicles in the network, by governing broadcasting in accordance with the vehicle's speed and direction, this technique leads to avoiding broadcasting an unnecessary HELLO beacon message, which in turn leads to improving the system's performance in terms of reducing packet collision and contention. The architecture has been designed based on the concept of a context awareness system, which has comprised three main stages; a collective knowledge phase, a processing phase and a reaction phase.

Moreover, the adaptive HELLO beacon message approach has been introduced as an alternative to control the process of sending a HELLO beacon message between vehicles, there are some situations that need to occur, in order to allow vehicles to broadcast their HELLO beacon messages, otherwise there is no need to broadcast them.

Chapter 5

Vehicle Second Heading Direction Routing Protocol (VSHDRP1)

Objectives:

- Defining the objectives behind developing the proposed routing protocol VSHDRP1
 - Describing the proposed routing protocol mechanism VSHDRP1 in a highway scenario using the aspect of the SHD
 - Implementing our proposed VSHDRP1
 - Describing the VSHDRP1 using a formal specification method
 - Evaluating the proposed VSHDRP1 using an NS-2 simulator
-

5.1 Introduction and motivation

In this chapter, a novel routing approach for the V2V type of communication in VANET is presented. Research into VANET has yielded considerable advances over the past few years, particularly in the areas of developing and designing new routing techniques [12]. Nevertheless, significant deficiencies of V2V ad hoc networks remain, especially when compared with V2I type of communications. In VANET, due to the expenses of installing an infrastructure that covers the road network, the V2I communication type is not feasible, when compared with the V2V communication type. Due to the mobility of these nodes, the probability of changing the interconnections between nodes can be significant. Therefore, the most important role for routing protocols in VANET is finding and maintaining robust and stable routes between a source node and a destination node. When designing a routing mechanism, the fundamental requirements that the routing algorithm has to meet are avoiding drop of the packet, making sure that the next-hop node has a stable trajectory and will travel on the current road for a specific period of time (choosing a long-lived route to the given destination) and requiring a small number of control messages to converge [47, 71].

As mentioned in chapter 3, serious impediments to mobility in VANET are the increasing rate of link failure in relation to the increasing mobility of nodes. This leads to an increase in both the congestion due to traffic backlogs and the volume of control traffic required to maintain routes. Thus, in order to achieve adaptive routing responsiveness and efficiency, the main goal when designing VSHDRP1 protocol is to reduce the reaction to mobility by increasing the route stability and packet delivery ratio.

The approach proposed in this chapter takes advantage of utilising the GPS system, NS and digital road map, in this way each vehicle can acquire knowledge of its position and direction, and can then establish a predetermined route, at the same time each vehicle can provide its neighbours with all this information by broadcasting a periodic HELLO beacon message. The proposed new protocol aims to initiate a robust and long-lived route between sources and destinations. This is in addition to reducing the flooding and overhead effects and minimising the rate of link breakages in the established paths. In the proposed approach, the decision to select nodes as the next hop node, to forward packets between source and the destination nodes is based on the Second Heading Direction (SHD) aspect of intermediate nodes, which represents the next road that the vehicle wants to take on its journey after following its current road. Several routing approaches have already been proposed with the aim of avoiding or reducing problems posed by the mobility of nodes in VANET, such as Vehicle - Assisted Data delivery Routing Protocol [12], Delay-Bounded Routing in Vehicular Ad-hoc Networks [50], Motion Vector Routing Algorithm (MOVE) [16], Scalable Knowledge-Based Routing (SKVR)[51], Static-node Assisted Adaptive Routing Protocol in Vehicular Networks (SADV) [52], Geographical Opportunistic Routing (GeOpps) [14] and MaxProp [53]. However not one of these routing protocol utilises the knowledge of the Second Heading Direction of each vehicle in the network.

The aim of the proposed routing protocol focuses on the process of delivering messages between vehicles from the source to its destination using a novel technique "Vehicle Second Heading Directional Routing Protocol" (VSHDRP1). In order to promise successful packet delivery to its destination; developing an

efficient routing mechanism is the main challenge that needs to be tackled; this is the main goal of this work. Our proposed project aims to enhance safety on the roads in an attempt to reduce the number of fatal accidents caused by automobile crashes. Employing wireless communication technology is considered to be a vital step towards improving safety applications in addition to other traditional safety measurements, such as airbag systems, which focus on reducing damage in the event of an accident.

However, applying wireless communication technology assists in preventing the occurrence of accidents, by warning vehicles located in a specific region about the possibilities of collisions occurring. In this way other vehicles approaching a dangerous situation can avoid accidents by reducing their speed or diverting to alternative route. For instance, this information could relate to road conditions ahead, the situation with the traffic in a specific area that a driver should know about, predicted weather conditions, or route a being blocked because of an accident, or route maintenance. This can be accomplished by exchanging information between vehicles via three alternative pathways: the first is performed through a coordinator which can be used to govern the activity involving exchanging information between vehicles creating a (V2I), using a Road Side Unit (RSU) [72, 73].

In contrast, the second alternative is based on exchanging information between vehicles directly (V2V). In these two alternatives, dedicated Short range Communication (DSRC) is used to perform the communications, based on the IEEE 1609 standards of Wireless Access in Vehicular Environments (WAVE) family [74, 75]. This introduces a homogenous interface between communicating vehicles. It comprises four standards; one of these standards is IEEE

1609.3, which is responsible for networking services including routing issues. A third alternative can involve using both types together; creating a Hybrid, for instance using the V2I when an infrastructure is available in a specific area that can be used by vehicles to exchange information; however, when this vehicle moves to another area where there is a lack of infrastructure vehicles can switch to the V2V communication type to exchange messages directly without the need for additional infrastructure.

5.2 VSHDRP1 Routing Protocol

In the proposed work, we intend to concentrate on using the V2V communication type solely without any type of infrastructure, owing to the high expense of installing different kinds of infrastructure from the perspectives of end users and companies; this, in addition to being unable to cover all the areas with infrastructures especially in rural areas, which are considered to be very expensive for companies [76]. We assume that each node has knowledge about its SHD, which can be acquired by utilising the Global Position System (GPS), Navigation System (NS) and preloaded digital maps; our proposed work depends mainly on this information, and on acquiring information about its surrounding neighbour nodes by exchanging periodic HELLO Beacon messages, which contains position, direction, speed and SHD information.

The main idea concerning the proposed routing protocol is that, the forwarding vehicle (which holds the packet) can select the next-hop vehicle (node) according to its SHD; if the next-hop node has a SHD that leads the vehicle to divert from its route at the next exit, then the path of the packet will continue ahead. In this situation the forwarding vehicle will ignore this vehicle and

look for an alternative vehicle that has SHD rather than leading the vehicle to the next exit. In other words, the forwarding vehicle will select the next-hop vehicle that continues in the same road without making any diversion in its route; thus the probability of delivering messages to a specific destination is increased (reduce the link breakage and the consequence will be increased system stability).

This proposed work mitigates Delay Tolerant Network (DTN) issues; it is based mainly on the carry-and forward strategy, when a disconnected area appears to be split between the vehicle that hold the packets and other vehicles that are moving towards the packet's destination; this causes a gap to occur between them and leads to the neighbouring node becoming unavailable to direct the node towards the desired destination. Using the knowledge of the vehicle's current position and the destination of the packet that needs to be reached is not sufficient.

Therefore, using the SHD is vital when deciding the next hop node. Suppose a forwarding node in a DTN Network needs to forward the packet to the next-hop node, which has the appropriate position and is directed towards the destination region, without knowing the SHD of the next hop, this next-hop node might take another route if it has a SHD leads the vehicle to another route before it reached its destination; in this case the SHD will have two values either 0 or 1, $SHD = 0$ means that the vehicle will not divert from its route at the next exit or intersection, while $SHD = 1$ suggests this vehicle will divert from its route at the next exit or intersection; in this situation this node is not suitable for delivering the packet, as a result the forwarding node will need to re-forward the packet, leading to reduced bandwidth consumption,

delay in packet transmission time and in system stability and connectivity.

In comparison the Greedy Perimeter Stateless Routing (GPSR) [46], is based mainly on position information from the neighbouring nodes without taking into account other parameters such as direction SHD or speed; in addition it does not utilise the aspect of carry-and-forward, instead when there is no neighbour node available with a closer position to destination node D , which means that the packet reach a local maximum. At this stage the greedy role of the GPSR mode is finished and the routing protocol switches to its other mode, the perimeter mode, used to maintain the block end. The concept of the perimeter mode in GPSR is to utilise the rule of right-hand with the starting vector constraint for the purpose of forwarding the packet.

Consequently, the next-hop in GPSR is found by following the right-hand rule (the next forwarding edge should be the first edge counter clockwise from the previous edge without crossing the starting vector). The packet switches back to the greedy forwarding mode in the process of forwarding the packet to a node that is closer to D ; otherwise it is dropped when a loop occurs. The drawback of the perimeter mode of GPSR is that it must be applied on a planar graph, or the crosslink may cause routing loops. GPSR introduces two schemes to apply to build a planar graph. However, issues such as obstacles and asymmetric radio range can prevent planar graphs from being created correctly.

This chapter introduced the process of the first anticipated VSHDRP1, which was designed for this research [71]. VSHDRP1 is a geographical routing proto-

col, which can be considered as operating in two parts: the first part manages the mobility knowledge acquired, whereas the second part is the packet sending process.

5.3 Mobility knowledge Acquisitions

In VSHDRP1 each mobile node sends information (e.g. position, velocity, current direction and SHD) to its surrounding nodes (neighbours) periodically. Therefore each vehicle in the network will acquire sufficient information regarding its neighbour nodes and the neighbours of the neighbour's nodes, in order to select the next hop node.

If a source node wants to send a packet to a destination, then it obtains knowledge of its own location, current direction, velocity and SHD through GPS, NS and preloaded road maps; in addition, the source node has sufficient knowledge about the position, velocity, current direction and SHD of each neighbouring node derived from periodic HELLO beacon messages, as shown below in table 5.1.

Field Name	Description
NT-type	Node type
Node-IP	Node IP Address
Node-dir1	Node First Direction (Current road direction)
Node-dir2	Node Second Heading Direction SHD (next road)
Node-pos	Node Current Position
Node-vlo	Node Moving Velocity
Nb	Information about its neighbour nodes

Table 5.1: HELLO beacon message format of VSHDRP1

5.4 VSHDRP1 Mechanism

This section describes the process of sending the packet from the source and intermediate node, and also the process of sending packet delivery confirmation.

5.4.1 Packet sending process

VSHDRP1 is based on the supposition that each vehicle (node) in the network has sufficient knowledge of its own location and direction, speed and SHD. As illustrated in flowchart in Figure 5.1, when a source node S needs to send a packet to a destination node D , it will look for the destination node in its cache (neighbours table), and if the destination node is found to be a neighbour in its cache, S will start forwarding the data packets to node D . If D is not found in the cache of source node S , then node S will set the current direction and the SHD, then it will start to look for an appropriate next-hop node using the filtration process for selecting the next-hop node as illustrated in flowchart in Figure 5.3, which comprises four main stages, which can be represented as: position stage, current direction stage, SHD stage and speed stage respectively. The output of the filtration process can be either $Yes(1)$ or $No(0)$.

- (a) *Yes*: This means an appropriate next-hop node is found from the neighbouring nodes, therefore the source node will forward the packet to this node
- (b) *No*: This represents the idea that no appropriate next-hop node is found from surrounding neighbour nodes. For that reason the source node will keep the packet in its buffer and continue listening for new neighbours to

be available. If a new neighbour node is found, the source node will run the same procedure until it delivers the packet to its destination.

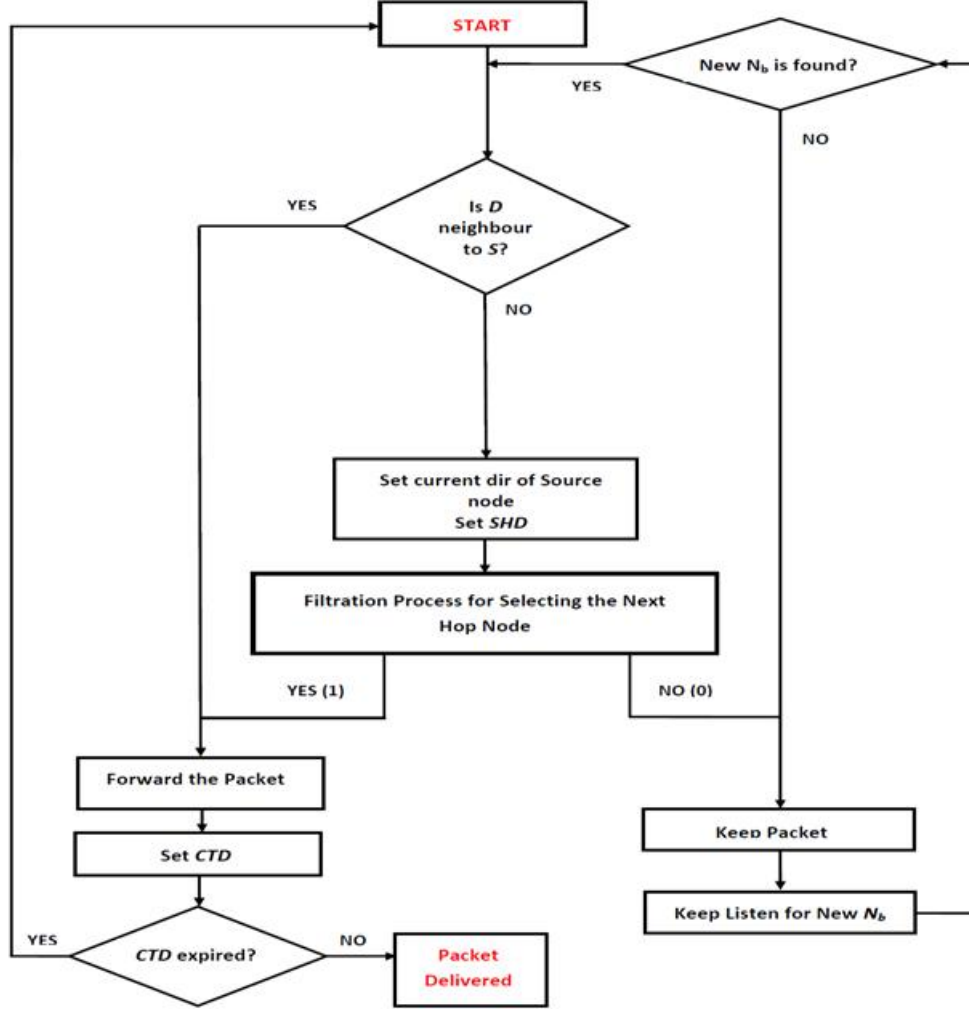


Figure 5.1: VSHDRP1 algorithm

5.4.2 Filtration process for selecting the next hop node

This section describes the four stages involved in selecting the next-hop node. In order to achieve the main target of this proposed protocol, which is represented by increasing the packet delivery ration and system stability Selecting the appropriate next hop node is a vital operation in this proposed protocol;

in this section it is necessary to clarify the operation for the four stages of the filtration process.

5.4.2.1 Filtration Process Stages

There are four stages for selecting the next-hop node as shown in Figure 5.2.

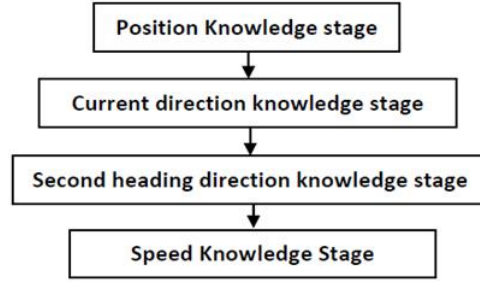


Figure 5.2: Filtration process diagram for selecting the next-hop node

- (a) Position Knowledge Stage: in this stage, the packet's holder node will select neighbour nodes that have a closer position to the packet's destination than itself.
- (b) Current Direction Knowledge Stage: in this stage, the selected nodes in the previous stage will be processed in order to check if they have an appropriate current direction (i.e. the direction towards the packet's destination).
- (c) Second Heading Direction Knowledge Stage (*SHD*): this stage nominates the candidates' nodes in the previous stage according to their *SHD*.
- (d) Speed knowledge stage: this stage will select the node with the highest speed in case more than one candidate are available.

5.4.2.2 Filtration Process Algorithm

As shown in Figure 5.3, the filtration process states that, if the destination node D is not found in the source node's cache, then the source node S will start the filtration process to look for an appropriate next-hop node. S will start the filtration process by looking for neighbours with a closer destination to D than itself. If no Neighbours' nodes (Nb) are found, it will buffer the packet; otherwise it will check if any of Nb nodes has a current direction towards the packet's destination. If an appropriate node is found then it will check to see if its $SHD = 0$; if more than one Nb node is found, then the Nb with highest speed will be selected. If $SHD = 1$ for all Nb , then S will check if $SHD = 0$ for a Neighbour of Neighbour Nb of Nb and if more than one Nb of Nb are found, then the Nb of Nb with the highest speed will be selected. If the current direction for all Nb is opposite to the packet's destination, then S will check the Nb of Nb that have current direction towards the packet's destination, and repeat the same procedure for SHD and speed.

5.4.3 Packet Delivery Confirmation

When the forwarding node intends to forward the packet to another intermediate node, it needs to make sure that the packet has been delivered successfully to that node; therefore it is important in this stage to send back an acknowledgment to the sending process.

Thus a time counter needs to be established at the same moment of sending the packet as shown in flowchart in figure 5.1; this counter is called Confirmation Time Duration (CTD). When the forwarding node wants to send the packet, it sets the CTD to a specified threshold value, and the value of this counter will be

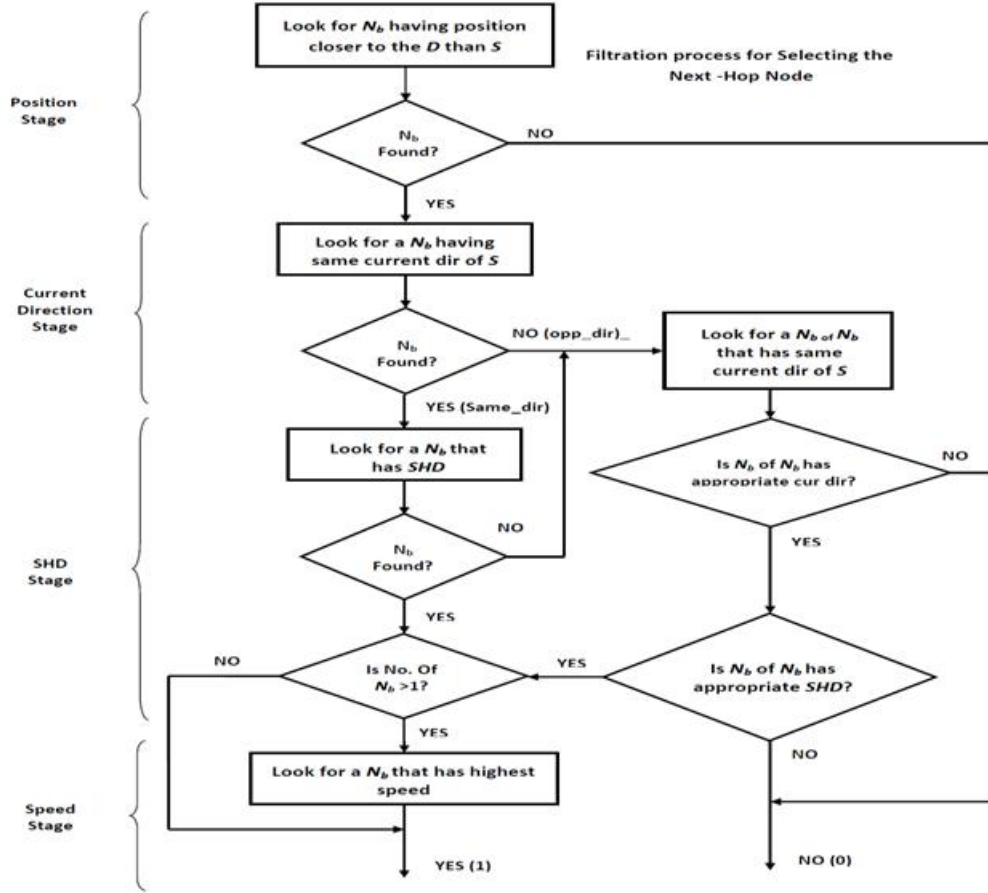


Figure 5.3: Filtration process algorithm for selecting next-hop node

decremented; if the confirmation is received before the CTD has elapsed, then the CTD will be halted. Otherwise, if the time is elapsed and no confirmation message (acknowledgment) has been received, that means the packet has been dropped or discarded for some reason; therefore the sender node will look for an alternative node and resend the packet.

5.5 Example of Packet Propagation

An example of the packet sending process at the source node using VSHDR protocol is showing below. As can be seen, S travel in a specific direction restricted by the road constraints, S need to send a packet to a specific desti-

nation, therefore it utilise the multi-hop technique, in figure 5.4, S will select the neighbour nodes that have position closer to the destination than itself, which means there is no necessity to select nodes like I_0 , because its position is not satisfying the condition of the first stage of filtration process, in the next stage S will check if those selected nodes have an appropriate current direction, as shown in figure 5.5, S will exclude node I_1 in case another nodes with an appropriate current direction are available such as nodes I_2 and I_3 , now S will check for the second heading direction of elected nodes (I_2 and I_3). The node with second heading direction equal to zero will be selected, which mean this vehicle will not divert its direction in the next exit or intersection, figure 5.6 show that if node S will fail in finding a neighbour node that have a closer position to destination than itself with a current direction equal to zero, then S will check for other neighbours with a current direction equal to one (opposite direction) that have neighbour nodes with a current direction equal to zero.

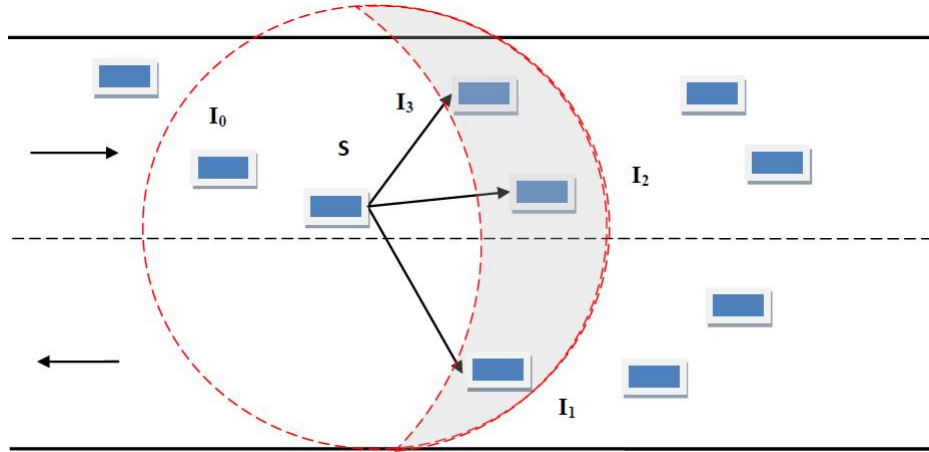


Figure 5.4: Next-hop node selection according to the neighbour's positions

Tables showing below illustrate that how the source node is selecting the next-hop node based on the status of each node, table 5.2 shows that each node in the network within its information (id , dir_1 , dir_2 and speed), according

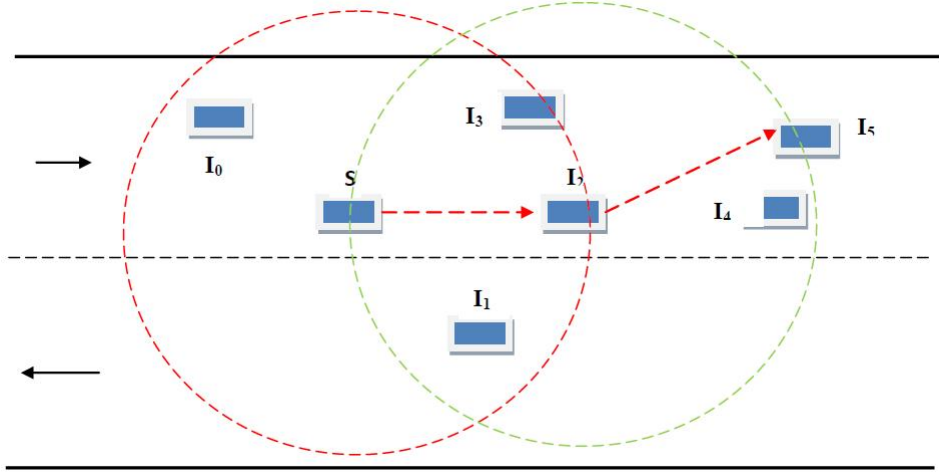


Figure 5.5: Next-hop node selection according to neighbour's current direction

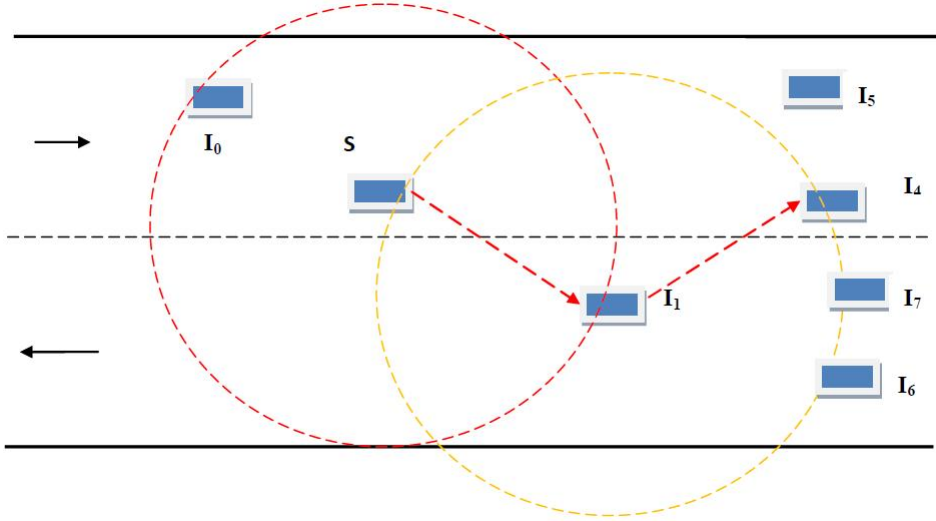


Figure 5.6: Next-hop node selection according to neighbour's opposite direction

to that information the route path of the packet is determined as shown in table 5.3 and 5.4 which represent the packet path followed in figure 5.5 and 5.6 respectively.

5.6 Formal Specification of VSHDRP1

We now give a formal specification of the VSHDRP1 protocol in CCA in a compositional manner, which is based on the example of the packet prorogation

node	Position	Dir1	Dir2	Velo
S	X,Y,Z	0	0	60m
I_0	X,Y,Z	0	0	50m
I_1	X,Y,Z	1	-	60m
I_2	X,Y,Z	0	0	60m
I_3	X,Y,Z	0	1	40m
I_4	X,Y,Z	0	0	70m
I_5	X,Y,Z	0	0	70m

Table 5.2: Information of nodes in the network

node	Position	Dir1	Dir2	Velo
S	X,Y,Z	0	0	60m
I_2	X,Y,Z	0	0	60m
I_5	X,Y,Z	0	0	70m

Table 5.3: Route path one

node	Position	Dir1	Dir2	Velo
S	X,Y,Z	0	0	60m
I_1	X,Y,Z	1	-	60m
I_4	X,Y,Z	0	0	70m

Table 5.4: Route path two

in section 5.5. Individual nodes in a VANET are specified independently and then composed in parallel to form the whole system.

5.6.1 System Model

A VANET is modelled in CCA as a parallel composition of all the nodes in the network (e.g. vehicles and road side units), i.e.

$$VANET \hat{=} node_0 \mid node_1 \mid \dots \mid node_{k-1} \quad (5.1)$$

Each node, $node_i$, in the VANET is modelled as an ambient of the following structure:

$$id[P_{id} \mid NH1[P_1] \mid NH2[P_2] \mid Hspeed[P_3] \mid Lspeed[P_4] \mid SHD[P_5]]$$

where

- id is the node's id. For the sake of simplicity, we use SN to denote the source node, DN for the destination node and IN_j for intermediate nodes, $j \geq 0$.
- P_{id} is a process that specifies the capabilities of the node, e.g. its ability to communicate or to sense the presence of other nodes in its range.
- $NH1$ is an ambient that contains the ids of the neighbouring node moving in the direction of the destination node (aka direction 1) and closer to the destination node.
- $NH2$ is an ambient that contains the ids of the neighbouring node moving in the direction opposite to direction 1 (this is called direction 2) and closer to the destination node.

- *Hspeed* is an ambient that contains the ids of the neighbouring nodes moving in high speed (i.e. speed greater than or equal to a specified threshold).
- *Lspeed* is an ambient that contains the ids of the neighbouring nodes moving in low speed (i.e. speed less than the threshold).
- *SHD* is an ambient that contains the ids of the neighbouring nodes with Second Heading Direction (i.e. SHD=0).
- Each process P_j , $1 \leq j \leq 5$, is either the inactivity process **0** or a parallel composition of ambients of the form $n[\mathbf{0}]$ where n is a node's id.

5.6.2 Context Expression

As explained in chapter 3, a context expression is a predicate that states the condition the environment of the executing process must meet. The context expressions used in the specification of the VSHDRP1 protocol are summarised as follows:

- $hasNb1(n)$ holds if the node n is a neighbour closer to destination in direction 1, i.e.

$$hasNb1(n) \triangleq \diamond(\bullet \mid NH1[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}).$$

- $hasNb2(n)$ holds if the node n is a neighbour closer to destination in direction 2, i.e.

$$hasNb2(n) \triangleq \diamond(\bullet \mid NH2[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}).$$

- $hasNb(n)$ holds if the node n is a neighbour closer to destination (regardless its direction), i.e.

$$hasNb(n) \triangleq hasNb1(n) \vee hasNb2(n).$$

- $noNb1()$ holds if there are no neighbours in direction 1, i.e.

$$noNb1() \triangleq \neg \diamond(\bullet \mid NH1[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True}).$$

- $hasSHD(n)$ holds if the node n is a neighbour with second heading direction (SHD), i.e.

$$hasSHD(n) \triangleq \diamond(\bullet \mid SHD[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}).$$

- $noSHD()$ holds if there are no neighbours with SHD:

$$noSHD() \triangleq \neg \diamond(\bullet \mid SHD[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True}).$$

- $hasNbofNb1(n)$ holds if node n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1:

$$hasNbofNb1(n) \triangleq hasNb2(n) \wedge \diamond(n[\mathbf{True} \mid NH1[\mathbf{True} \mid \oplus \mathbf{True}]] \mid \mathbf{True}).$$

- $highestSpeed(n)$ holds if the node n has the highest speed:

$$\begin{aligned} highestSpeed(n) \triangleq & \\ & \diamond(\bullet \mid Hspeed[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}) \vee \\ & (\neg \diamond(\bullet \mid Hspeed[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True}) \wedge \\ & \diamond(\bullet \mid Lspeed[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})) \end{aligned}$$

We now specify the behaviours proper to each type of node (source node, intermediate node and destination node).

5.6.3 Source Node

A source node SN is the node that initiates a run of the VSHDRP protocol, willing to send a message msg to a destination node DN . So its capabilities are modelled by the following process:

$$P_{SN} \triangleq (\nu x) \{Eq.5.3 \mid Eq.5.4 \mid Eq.5.5 \mid x[\uparrow send().0]\} \quad (5.2)$$

Where the restricted name x is used to guarantee that not more than one of the processes Eq.5.3, Eq.5.4 and Eq.5.5 are executed. Indeed, x is an ambient

that sends a single signal to its parent; this signal will be captured by exactly one of these processes, eventually. These processes are specified as follows:

- if the destination node is a neighbour, send message to destination node and wait for acknowledgement. This is formalised as:

$$hasNb(DN)?x \downarrow \text{recv}().DN :: \text{send}(SN, msg).DN :: \text{recv}(y).0 \quad (5.3)$$

- if the destination node is not a neighbour then look for an intermediate node moving in direction 1 with SHD and highest speed, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left. \begin{array}{l} \text{find } n : E1(n, DN) \text{ for } x \downarrow \text{recv}().n :: \text{send}(SN, DN, msg). \\ n :: \text{recv}(y).0 \end{array} \right\} \quad (5.4)$$

where

$$E1(s, t) \triangleq \neg hasNb(t) \wedge hasNb1(s) \wedge hasSHD(s) \wedge highestSpeed(s)$$

- if no such intermediate nodes then look for an intermediate node moving in direction 2 which has a neighbour in direction 1 closer to the destination node, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left. \begin{array}{l} \text{find } n : E2(n, DN) \text{ for } x \downarrow \text{recv}().n :: \text{send}(SN, DN, msg). \\ n :: \text{recv}(y).0 \end{array} \right\} \quad (5.5)$$

where

$$E2(s, t) \triangleq \neg hasNb(t) \wedge (noNb1() \vee noSHD()) \wedge hasNb1ofNb1(s)$$

5.6.4 Intermediate Node

An intermediate node IN receives a triple $(sender, dest, msg)$ where $sender$ is the sender's id, $dest$ is the destination node's id and msg is the message being sent. The intermediate node confirms the receipt by sending an acknowledgement to the sender and forwards the message to an appropriate node. This is specified as:

$$P_{IN} \triangleq \left. \begin{array}{l} ! :: \mathbf{recv}(sender, dest, msg).sender :: \mathbf{send}(ack). \\ (\nu x) \{Eq.5.7 \mid Eq.5.8 \mid Eq.5.9 \mid x[\uparrow \mathbf{send}().0]\} \end{array} \right\} \quad (5.6)$$

where the restricted name x plays the same role as in Eq.5.2 for selecting at most one of the processes Eq.5.7, Eq.5.8 and Eq.5.9. The replication operator ‘!’ means that an intermediate node repeats this pattern of behaviour its whole lifetime. Moreover, an intermediate node determines the next node to forward the message to as follows:

- if the destination node $dest$ is a neighbour, send message to destination node and wait for acknowledgement, i.e.

$$hasNb(dest)?x \downarrow \mathbf{recv}().dest :: \mathbf{send}(IN, msg).dest :: \mathbf{recv}(y).0 \quad (5.7)$$

- if the destination node $dest$ is not a neighbour then look for another intermediate node moving in direction 1 with SHD and highest speed, and send the message to that intermediate and wait for acknowledgement, viz.

$$\left. \begin{array}{l} \mathbf{find} \ n : E1(n, dest) \ \mathbf{for} \ x \downarrow \mathbf{recv}().n :: \mathbf{send}(IN, dest, msg). \\ n :: \mathbf{recv}(y).0 \end{array} \right\} \quad (5.8)$$

- if no such intermediate nodes then look for another intermediate node moving in direction 2 which has a neighbour in direction 1 closer to the

destination node, and send the message to that intermediate and wait for acknowledgement, viz.

$$\left. \begin{array}{l} \text{find } n : E2(n, dest) \text{ for } x \downarrow \text{recv}().n :: \text{send}(IN, dest, msg). \\ n :: \text{recv}(y).0 \end{array} \right\} \quad (5.9)$$

5.6.5 Destination Node

The destination node DN receives a pair $(sender, msg)$ and sends an acknowledgement to the sender. This behaviour is formalised as:

$$P_{DN} \hat{=} :: \text{recv}(sender, msg).sender :: \text{send}(ack).0 \quad (5.10)$$

This formal specification of the VSHDRP1 protocol is executable by the CCA interpreter and is used in the following section to simulate runs of the protocol.

5.7 Behavioural Validation and Analysis based on CCA

In this section, we run the formal specification of the VSHDRP protocol presented above in a variety of scenarios reflecting differing configurations of the VANET network. In these scenarios, a source node SN is willing to send a “hello” message to a destination node DN . Other nodes of the network are named IN_1 , IN_2 and so on. The result of each run of the protocol is a sequence of reductions showing the interactions that happened among the VANET’s nodes. This output can then be analysed to check that properties of the protocol are preserved, at an early stage prior to implementation and de-

ployment. The full formal executable specification of VSHDRP1 is illustrated in Appendix B. For illustration, we consider three scenarios as follows.

5.7.1 Scenario 1

The destination node is neighbour to the source node as shown in Figure 5.7. This scenario is set up by adding the *id* of the destination node in NH1 component of the source node and the *id* of the source node in the NH1 component of the destination node. The output of the run is depicted in Table 5.5, where line numbers are added to make it easy to refer to individual lines.

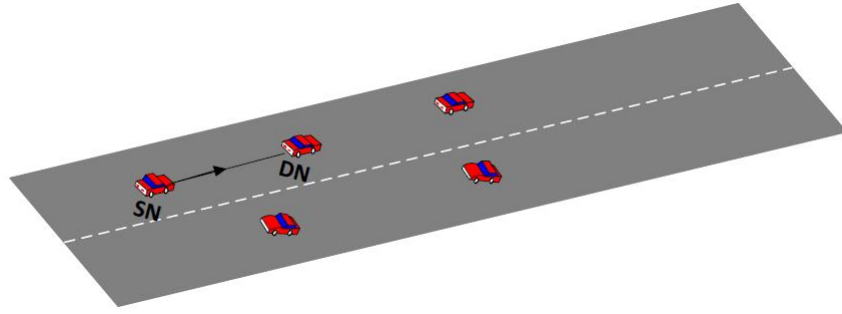


Figure 5.7: CCA Scenario1

1.	$\langle \rightarrow \rangle$ {renaming of a restricted name: x to $x\$0$ }
2.	\rightarrow {Child to parent: $x\$0 ==() \Rightarrow SN$ }
3.	\rightarrow {Sibling to sibling: $SN == (SN, \text{hello}) \Rightarrow DN$ }
4.	\rightarrow {Sibling to sibling: $DN == (\text{ack}) \Rightarrow SN$ }

Table 5.5: Output of CCA scenario 1

The symbol ' $\langle \rightarrow \rangle$ ' corresponds to the structural congruence relation which is an equivalence relation between processes, while ' \rightarrow ' represents the reduction relation of CCA which determines a process transition as formally defined in [8]. The explanation of each transition is given between a pair of curly brackets. In particular, the notation ' $A == X \Rightarrow B$ ' means that an ambient ' A '

sent a message ‘X’ to another ambient ‘B’. Other notations such as ‘Child to parent’ and ‘Sibling to sibling’ provide information about the relationship between the sender ‘A’ and the receiver ‘B’.

The line 1 in Table 5.5 corresponds to the execution of the restriction (νx) in Eq. 5.2 which renames the name x to a new name $x\$0$. In line 2, the source node SN detects that the destination node DN is a neighbour and decides to execute the process in Eq. 5.3 by receiving a signal from the ambient $x\$0$. Then the “hello” message is sent directly to the destination node (line 3) which acknowledges in line 4 and the run terminates.

5.7.2 Scenario 2

In this scenario, the destination node is not neighbour to the source node; rather the following neighborhood relationship is considered: $\{(SN, IN_1), (IN_1, IN_2), (IN_2, IN_3), (IN_3, DN)\}$ as shown in Figure 5.8. In addition, the node IN_2 moves in direction 1, while IN_1 and IN_3 move in direction 2. Each of these nodes has a second heading direction (SHD). The output of the protocol’s run is given in Table 5.6.

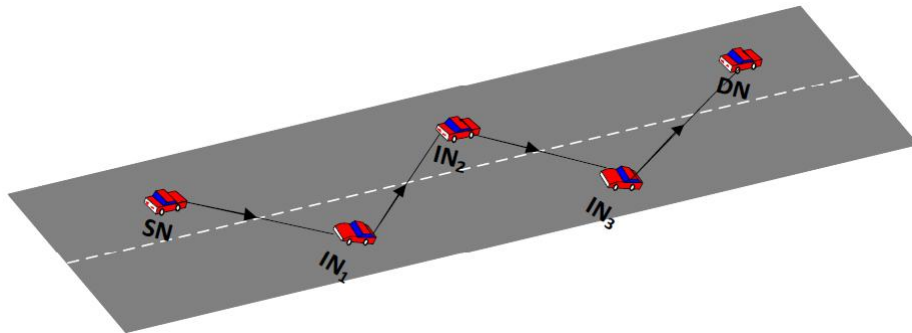


Figure 5.8: CCA Scenario 2

```

1. <--> {renaming of a restricted name: x to x$0}
2. --> {binding: n -> IN1}
3. --> {Child to parent: x$0 ==()=> SN}
4. --> {Sibling to sibling: SN ==(SN,DN,hello)==> IN1}
5. --> {Sibling to sibling: IN1 ==(ack)==> SN}
6. <--> {renaming of a restricted name: x to x$11}
7. --> {binding: n -> IN2}
8. --> {Child to parent: x$11 ==()=> IN1}
9. --> {Sibling to sibling: IN1 ==(IN1,DN,hello)==> IN2}
10.--> {Sibling to sibling: IN2 ==(ack)==> IN1}
11.<--> {renaming of a restricted name: x to x$20}
12.--> {binding: n -> IN3}
13.--> {Child to parent: x$20 ==()=> IN2}
14.--> {Sibling to sibling: IN2 ==(IN2,DN,hello)==> IN3}
15.--> {Sibling to sibling: IN3 ==(ack)==> IN2}
16.<--> {renaming of a restricted name: x to x$29}
17.--> {Child to parent: x$29 ==()=> IN3}
18.--> {Sibling to sibling: IN3 ==(IN3,hello)==> DN}
19.--> {Sibling to sibling: DN ==(ack)==> IN3}

```

Table 5.6: Output of CCA scenario 2

In line 2 the notation `{binding: n -> IN1}` corresponds to the execution of a *search prefix* and means that the node IN1 is found (by the source node SN) as an appropriate node to forward the message to. The source node SN then forwards the message to the node IN1 in line 4. In a similar way, the message goes from IN1 to IN2 then to IN3 and finally to DN as showed in lines 9, 14 and 18 respectively.

5.7.3 Scenario 3

In this scenario, there is a disconnected area between the source node and destination node. This is modelled by the following neighborhood relationship: $\{ (SN, IN1), (IN1, IN2), (IN3, DN) \}$.

As shown in Figure 5.9, the disconnected area is between the node IN2 and the node IN3 as there is no routes that link them. The output of the protocol's run in this case is given in Table 5.7.

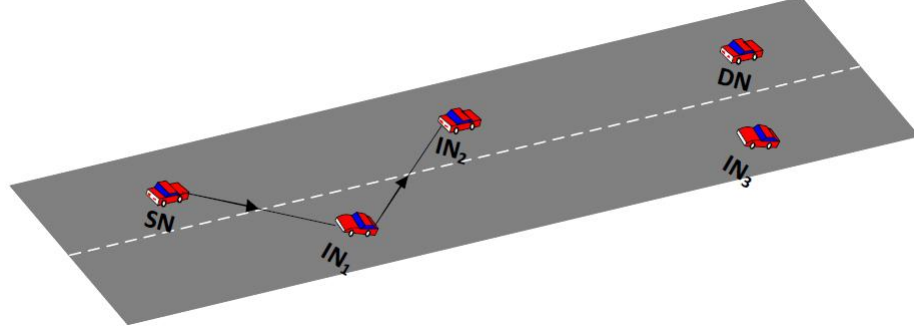


Figure 5.9: CCA Scenario3

```

1.<---> {renaming of a restricted name: x to x$0}
2.---> {binding: n -> IN1}
3.---> {Child to parent: x$0 ==()==> SN}
4.---> {Sibling to sibling: SN ===(SN,DN,hello)==> IN1}
5.---> {Sibling to sibling: IN1 ===(ack)==> SN}
6.<---> {renaming of a restricted name: x to x$11}
7.---> {binding: n -> IN2}
8.---> {Child to parent: x$11 ==()==> IN1}
9.---> {Sibling to sibling: IN1 ===(IN1,DN,hello)==> IN2}
10.---> {Sibling to sibling: IN2 ===(ack)==> IN1}
11.<---> {renaming of a restricted name: x to x$20}

```

Table 5.7: Output of CCA scenario 3

It follows that the message goes from the source node SN to the node IN1, then to IN2 and gets stuck at this node. However, the node IN2 will forward the message as soon as connection is established.

5.8 Straight Highway Scenario

This highway case study shows the situation of a safety requirements, as an accident occurring over the road, where there are no enough vehicles in the road to acts as an intermediate node to forward the packets to its exact destination (in sparse environment), this destination can be an ambulance vehicle. The VSHDRP1 will be the perfect solution to such a scenario. As known the sparse environment is quite challenged, particularly when there is a critical need to deliver a packet to warn the nearest ambulance vehicle about the occurrence of the accident.

Assume an accident occurs on a highway, as shown in 5.10; node Z sends a warning message to the ambulance vehicle (Amb), that an accident has occurred, as well as to advise other vehicles to avoid the hazardous situation (jam). At the same time, there is no vehicle available in zone X to alert vehicles approaching the accident location in order to adjust their speed or take another route as an attempt to avoid crashes. This absence of vehicles in zone X makes a disconnected (sparse environment) area between the accident location and zone Y . Therefore, the only way to inform the vehicles in the zone y (including the Ambulance vehicle Amb) that an accident has happened (if we assume there are no infrastructures over the road) is through the cars on the other side (direction) of the road; node S receives the warning message from node Z , and node S now has to forward the message to other vehicles in the road. In this case, this vehicle will use the store-and-forward technique owing to the lack of nodes in zone G . However, what would happen if just one of these node reaches zone Y and other vehicles divert to the right direction. Another assumption in this scenario can arise: if node S forwards the message

to all vehicles moving in the desired direction, but not all of these vehicles complete their journey to the end of the road (take the exit), that will lead to the packet being dropped and vehicles in zone Y not being alerted. Thus, it is useful to use the information of the second direction road of each vehicle, to enable the forwarding vehicle to take the right definition in forwarding the packet, because if no vehicle reaches the desired destination, the forwarding vehicle will keep the packet until it obtains an appropriate next hop node. In other words, there is no need to forward the message to other vehicles if we know that they will not reach the destination area (zone Y). This scenario can be achieved through using additional information in the beacons, such as Second Heading Direction (SHD); that can be provided by using external devices such as road maps and navigation systems. Applying this method leads to the use of the bandwidth being organised in an ideal way; consequently, not all nodes in zone Y need to receive this warning message if they change their direction before reaching accident location.

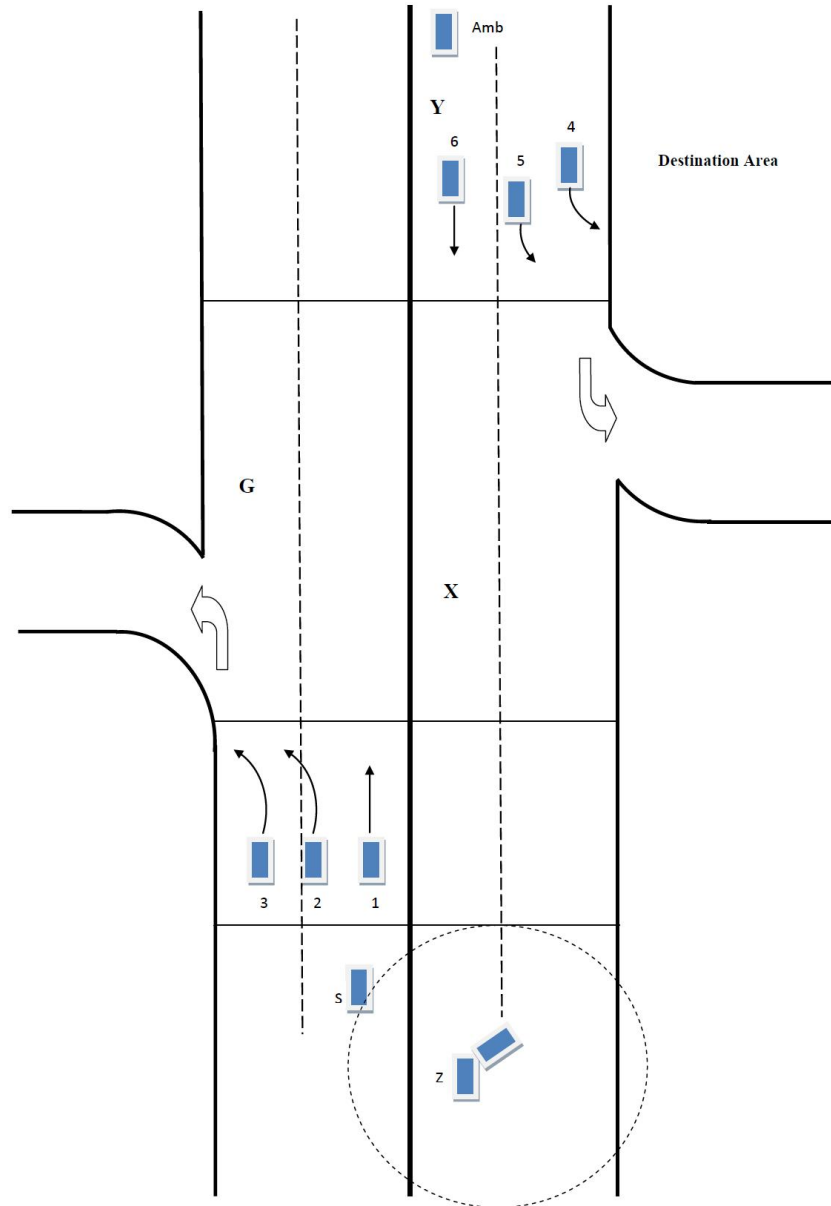


Figure 5.10: Highway Scenario

5.9 Simulation Methodology and Model

There are two main reasons for the nodes behaviour being unknown, the first reason is the obscurity associated with predicting a context for VANET behaviour. The second one is that VANET is considered mainly as a research

subject. Thus it is extremely costly and complicated to take authentic measurements in the field. Hence VANET routing protocols are typically evaluated and compared with each other using simulation tools. El-Nabi [77] explained this by saying: *"There are three different ways to model networks: formal analysis, real life measurements and simulation [78]. The dynamic nature of ad hoc networks makes them hard to study by formal analysis. Some formal techniques that have been used in static networks include Petri nets, stochastic processes, queuing theory, and graph theory. None of these is especially well suited to studying dynamic networks. Since ad hoc networks are still mainly a research subject, most scenarios they will be used in are still unknown. For those scenarios that are known, e.g. military networks, extreme uncertainties and dynamicity are expected. Thus, use of real life measurements is currently almost impossible and certainly costly. The commonly used alternative is study the behaviour of the protocols in a simulated environment"*.

Additionally, the efficiency of network behaviours can be experienced and evaluated by using a simulation, even when there are a high number of mobile nodes and a variety of network area sizes, which provide more flexibility than physical devices in the case of a fixed area. Furthermore, it provides flexible control over the values of protocol parameters and iteration running, by varying one set of variables and setting the remainder as constants; this leads to the promise of a near ideal way to determine the network parameters.

5.10 Simulation Environment

Version 2.34 of NS-2 has been used to simulate the VSHDRP1 mechanism, the NS-2 has been installed in a Linux-based operating system Ubuntu 10.4. The

VSHDRP1 has been simulated so as to be evaluated and compared with other routing protocols. The reason behind selecting the NS-2 simulator for this work, is that it provides a range of characteristics that make it a distinguished simulator; NS-2 is an open source code simulator that can be easily modified and expanded.

5.11 Parameter Values

In this simulation model, the number of mobile vehicles is set to range from 10 to 150 mobile vehicles placed randomly within the boundaries of the simulation area restricted by travel in only two directions, so as to represent the two opposite directions on the road. Each simulation was executed for 250 seconds of simulation time. The deploy area for each simulation was chosen as a 4000m x 3200m rectangle area, representing a street. The constant Bit Rate (CBR) is set to vary from 0.1 to 1.0 as a network traffic model. The pattern of communications used in the simulation is peer-to-peer. The speed of vehicles in this simulation varies from 20-75 *mph*, all the settings given above are intended to reflect a near realistic situation of a road layout in an authentic scenario. The performance of the protocol is evaluated using a set of metrics; these are data delivery ratio, generated traffic overheads and data-delivery delay. All parameters settings are shown in table 5.8.

Parameters	Network Size Scenario	Data Packet Size Scenario	Vehicle beacon interval Scenario
No. of Nodes	10, 20, 50, 100, 150	10	10
Max. Vehicle Speed	25-75 m	25-75 m	25-75 m
Simulation Time	250 sec	250 sec	250 sec
Network Space	4000X3200 meter	4000X3200 meter	4000X3200 meter
Radio range	250 meter	250 meter	250 meter
MAC Protocol	IEEE 802.11	IEEE 802.11	IEEE 802.11
Radio Propagation Model	Free Space/ Two Ray	Free Space/ Two Ray	Free Space/ Two Ray
Data Packet Size	10b	10 b - 4 Kb	10b
Antenna Model	Omni Antenna	Omni Antenna	Omni Antenna
Traffic Pattern	CBR	CBR	CBR
Vehicle Beacon Interval	5.0 sec	5.0 sec	0.2 sec, 0.5 sec, 1.0 sec, 5.0 sec

Table 5.8: NS-2 simulation Setup

5.12 The Result and Analysis of VSHDRP1 Routing Protocol simulation

As mentioned previously, the simulation result of the VSHDRP1 is obtained through the NS-2 simulator version (2.34), in order to make an evaluation of the performance of this routing protocol, based on appropriate metrics, we compare the simulation result from the VSHDRP1 with another routing protocol GPSR, GPSR is considered to be one of the standard routing protocols in ad hoc networks and VANET.

In order to obtain a fair comparison, we re-simulate the GPSR routing protocol in the same environment as that used to simulate the VSHDRP1, and show the comparison results based on performance metrics.

5.13 Simulation Metrics

The simulation result has been analysed by comparing three different metrics to evaluate the performance of the routing protocol, these metrics are as

follows:

- *Data packet delivery ratio*: representing the number of data packets delivered to the destination successfully of all the packets sent in the network, including the packet forwarded during transmission. The goal when using this metric is to measure the efficiency of the data packet delivery.
- *Overhead*: representing the total number of control packets that are sent during the processing of delivering the data to its destination.
- *End-To-End Delay*: represents the time delay needed to transfer a packet from the source to its destination, including the time consumed during the process of buffering and retransmission operations.

5.14 Result and Analysis

In this section the performance of VSHDRP1 was evaluated in terms of the efficiency of data packet delivery ratio, number of packets sent through the network (overhead) and time delay.

5.14.1 The Efficiency of Data Packet Delivery Ratio

In this section the efficiency of the data packet delivery ratio for the VSHDRP1 has been measured against three scenarios; network size, data packet size and HELLO beacon message period intervals. The measure of efficiency obtained was then compared with the GPSR, the performance of which was also evaluated based on the same scenarios.

5.14.1.1 Network Size

Here we will compare the performance of the VSHDRP1 with the GPSR in terms of the data packet delivery ratio to test the impact of a varying number of vehicles and the vehicular interval on the measurements. The chart in Figure 5.11 represents a comparison to show the efficiency of the data packet delivery ratio, versus increasing the number of nodes (network size). Firstly, the chart shows a noticeable increase in the efficiency of data packet delivery ratio when there is an increasing number of nodes in the network, especially in the left hand side of the graph as this protocol VSHDRP1 is more concerned about low density network; the reason for which is, the more nodes that exist the fewer the disconnected areas between nodes, a factor that leads to more successful transmission between source nodes, intermediary nodes and ultimately the destination node.

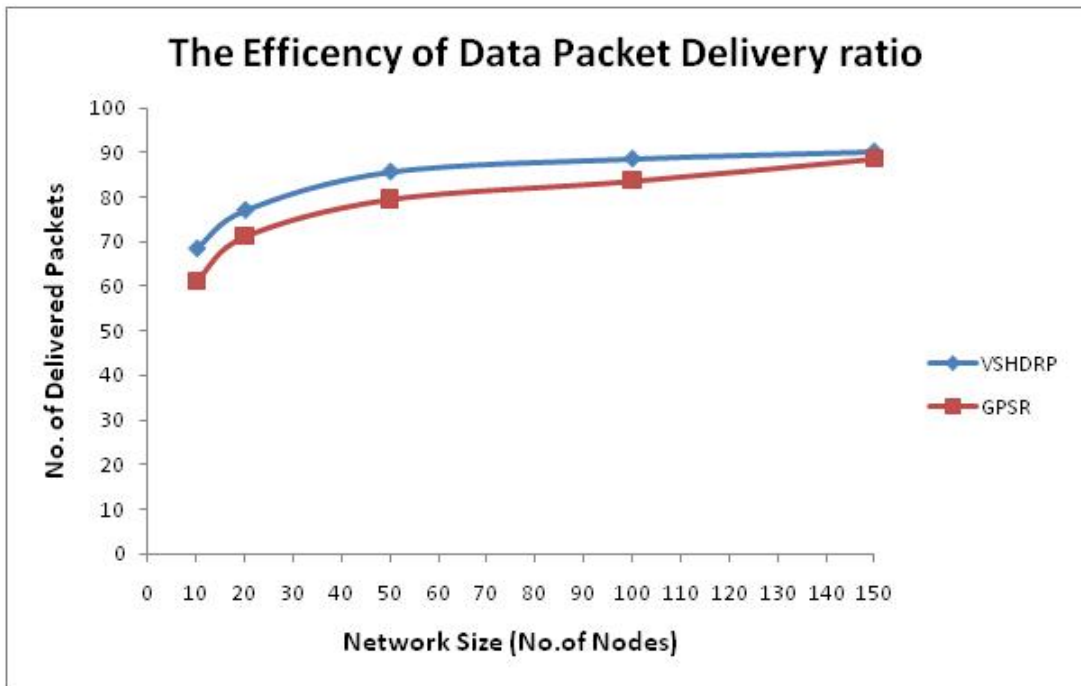


Figure 5.11: The efficiency of data packet delivery against the nodes number (network size)

Furthermore; the efficiency of data packet delivery ratio with VSHDRP is better than the efficiency of data packet delivery ratio with GPSR even with a low density of nodes (DTN network); the reason being, that VSHDRP1 exploits SHD which increases the route link stability and reduces the possibility of packet drop through the selection of the right node as intermediary, in order to reliably forward the packet to its destination. In addition, it shows that the VSHDRP1 is able to provide better data packet delivery ratio than GPSR, even with low number of nodes, which achieves the designers aim in developing this protocol.

5.14.1.2 Data Packet Size

The chart in Figure 5.12 represents the efficiency of data packet delivery ratio versus an increase in the size of the data packet; as we can see from the figure, the efficiency of data packet delivery ratio shows a slight decrease, which then becomes settled with the increase in data packet size for the same number of packets. The reason for this is that, the data packet size has more impact on the bandwidth consumption, which are considered to be limited in VANET, that causes an issue for contention to the wireless channels, .

In addition, however, the chart shows a significant improvement in VSHDRP1 performance in terms of the efficiency of the data packet delivery ratio over GPSR, due to the exploitation of the aspects of SHD in the process of filtering neighbours to select the appropriate next-hop node to forward the packet to its destination.

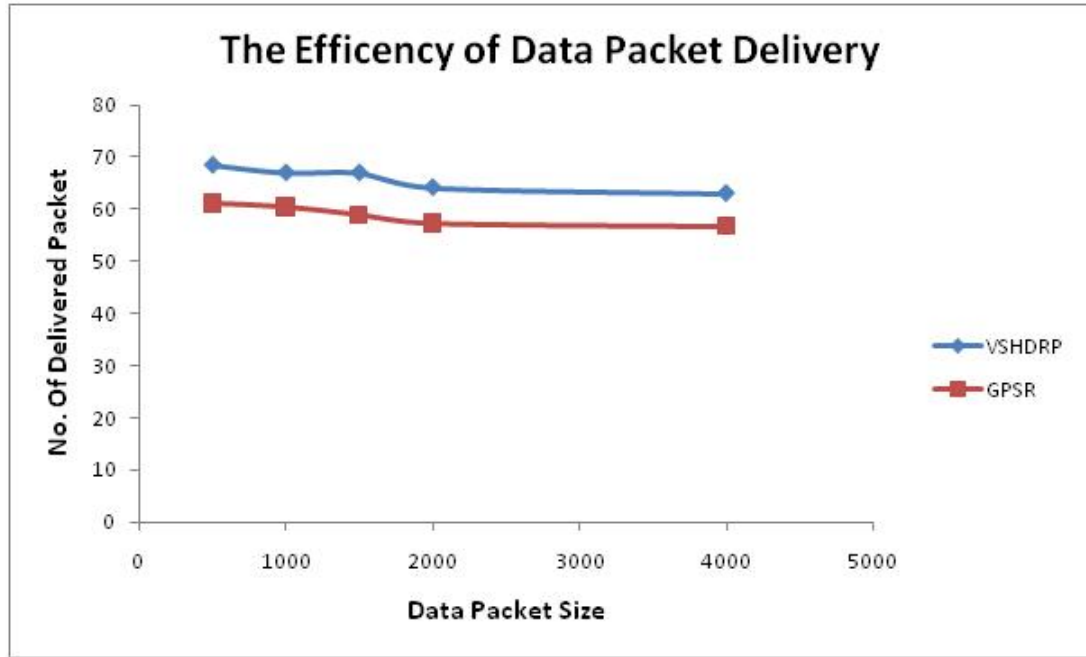


Figure 5.12: The efficiency of data packet delivery against the data packet size

5.14.1.3 HELLO beacon Message Period Intervals

The chart in Figure 5.13 represents the efficiency of the data packet delivery ratio versus an increase in the periodic interval of broadcasting the HELLO beacon message, in general. The chart shows a significant decrease, especially initially; it then starts to settle due to the increasing time interval in broadcasting the HELLO beacon message that affects the nodes in the network by preventing them from being up-to date with the other nodes in the networks, which in turn leads to them forwarding the packets to an improper node, which may lead to the packet being dropped, impacting the delivery ratio. In addition, the chart shows a significant improvement in the performance of VSHDRP in terms of the efficiency of the data packet delivery ratio over GPSR.

A significant decrease in the beginning of the chart is considered to be a

consistency in all charts of varying the HELLO beacon messages intervals 5.13, 5.16 and 5.19. where the interval of broadcasting HELLO beacon message is considered to be less in beginning area of these three charts. Here nodes have an up to date information about their neighbours nodes, for that reason they show a significant changes in all the three charts.

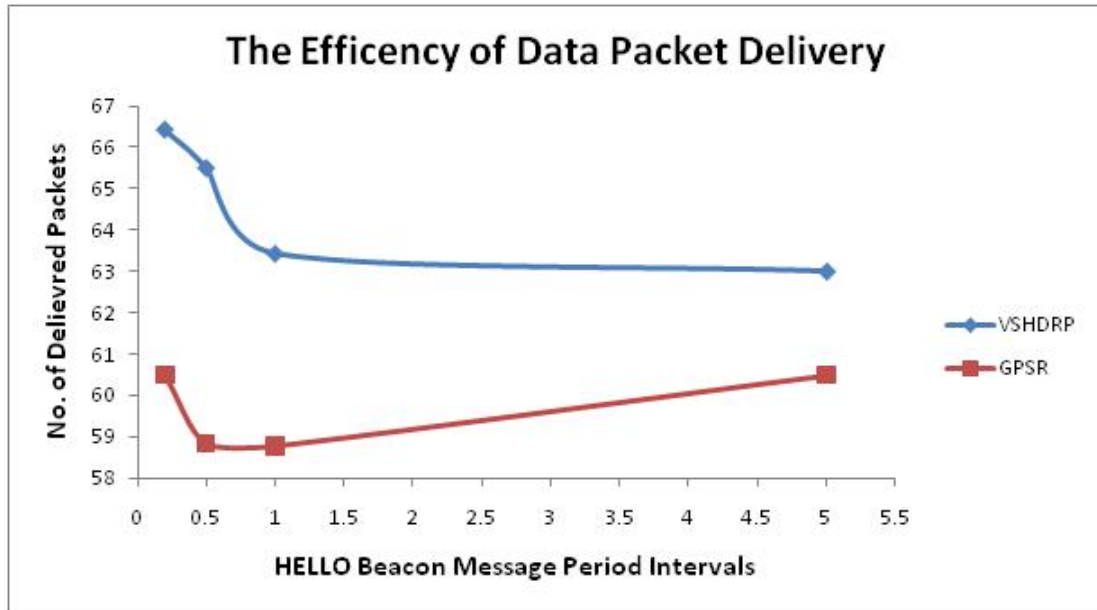


Figure 5.13: The efficiency of data packet delivery against the beacon message intervals

5.14.2 Generated Data Traffic (Overhead)

In this section the Number of Packets Sent through the Network (Overhead) for the VSHDRP1 has been measured against three scenarios network size, data packet size and HELLO beacon message period intervals, and compared with the GPSR which its performance has been evaluated based on the same scenarios.

5.14.2.1 Network Size

The chart in Figure 5.14 depicts the number of packets sent through the network (Overhead) as compared to the variety of network sizes; it shows that the overhead is increased when increasing the number of nodes in the network, because the increase in the number of nodes leads to enlargement of the overhead packet produced through the network. In addition; it shows that the VSHDRP produced less overhead than the GPSR, which means that the performance of the VSHDRP is better than the GPSR, because the reduction that occurs results in link breakage, reducing the number of control packets that need to be sent though the network to find alternative routes to deliver the packet to its destination.

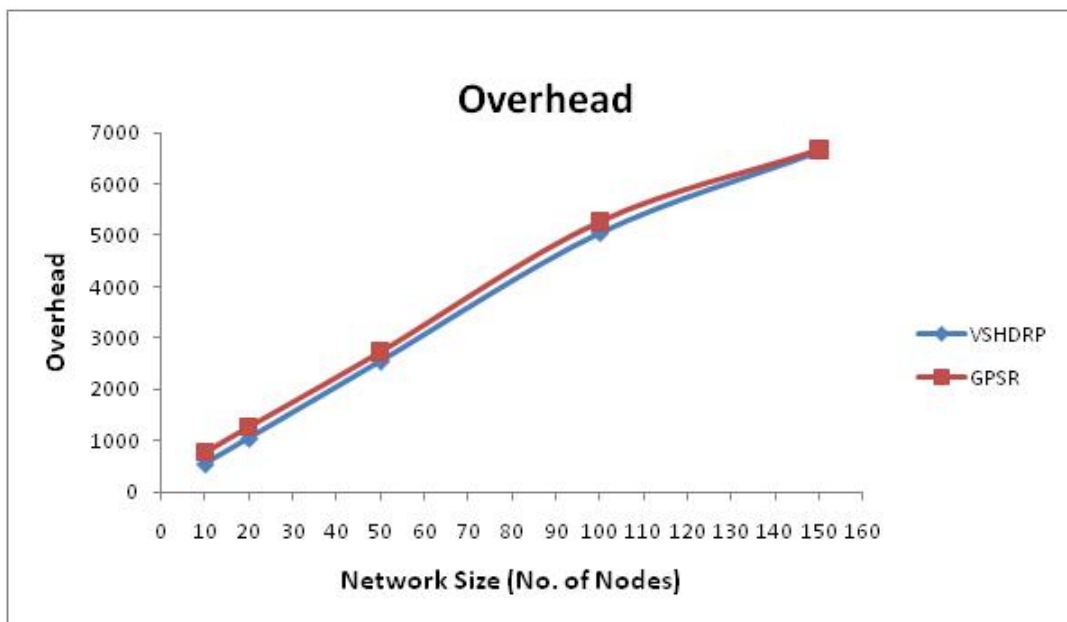


Figure 5.14: Overhead against network size

5.14.2.2 Data Packet Size

The chart in figure 5.15 illustrates the impact of varying the data packet size on the number of packets sent through the network (Overhead). It reveals that the overhead is settled by increasing the data packet size, because the data packet size has more impact on bandwidth consumption, which causes an issue of contention when using the wireless channels.

Additionally, the chart shows a significantly better performance for VSHDRP1 over the GPSR. This derives from the fact that it utilised the aspect of SHD in the filtration process, in order to select the next-hop node effectively; involving moving towards the desired destination, leading to a reduction in failed attempts to deliver the packet to its destination, which has an impact on reducing the overheads.

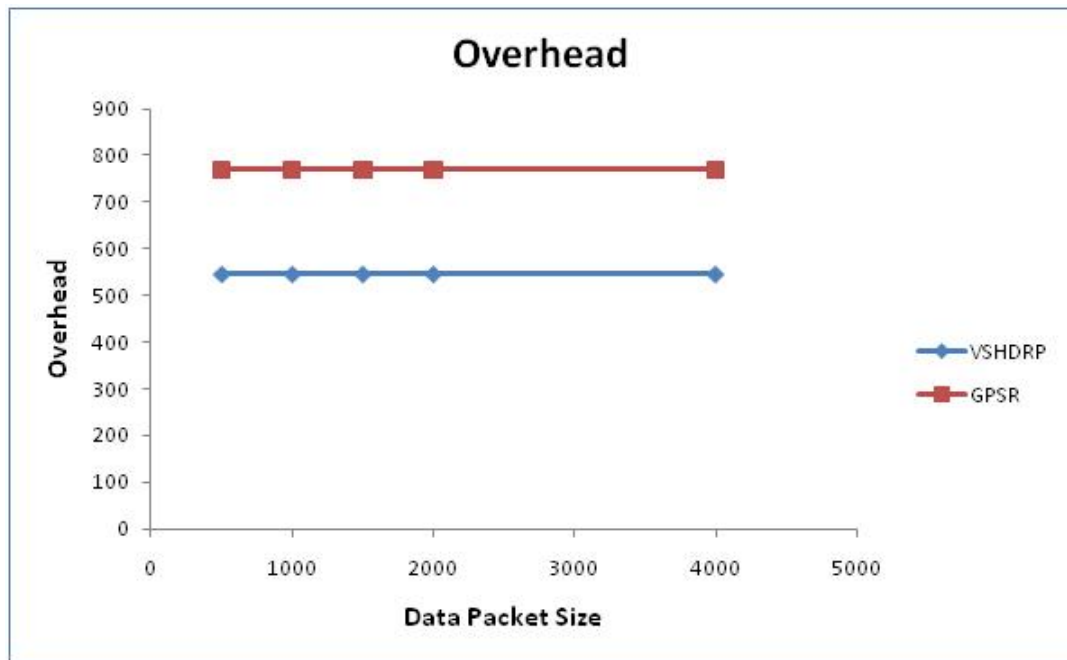


Figure 5.15: The data packet size against the number of packets

5.14.2.3 HELLO beacon Message Period Intervals

The chart in Figure 5.16 illustrates that the impact of varying the period time interval between broadcasting the HELLO beacon message on the number of packets sent through the network (Overhead), showing that the overhead is decreased significantly when increasing the time interval, because this reduces the number of HELLO messages per time unit which means fewer control packets are sent. In general VSHDRP shows less overhead than GPSR, principally due to the presence of SHD in the filtration process to select the next-hop node effectively; this moves towards the desired destination reducing the number of failed attempts to deliver the packet to its destination.

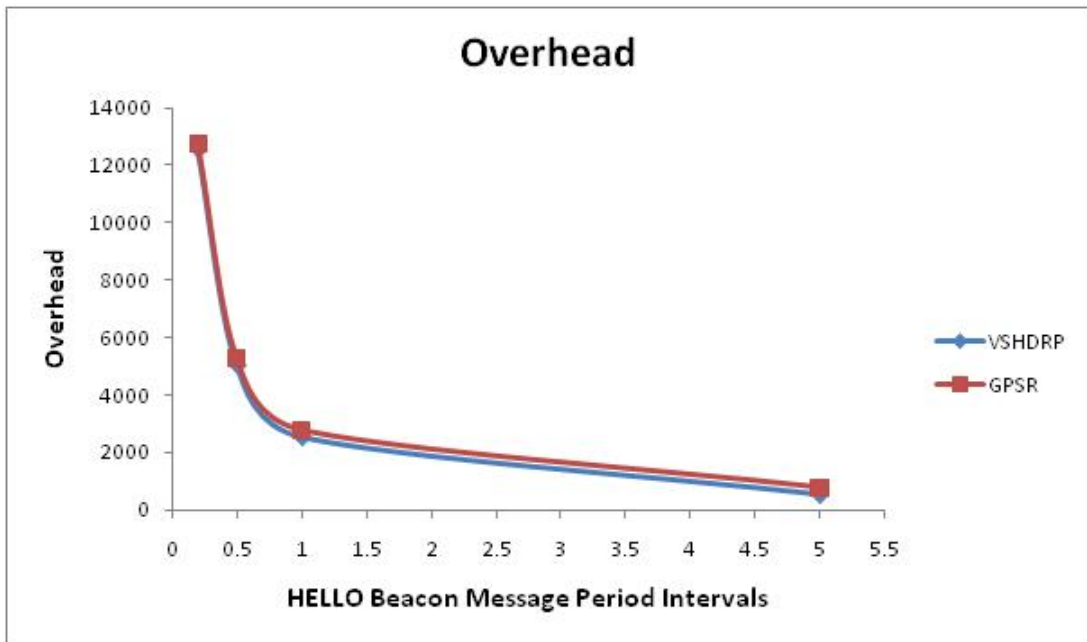


Figure 5.16: Overhead against the beacon message intervals

5.14.3 The Time Delay of Data Delivery

In this section the End-To-End Time Delay for the VSHDRP1 has been measured against three scenarios network size, data packet size and HELLO beacon

message period intervals, and compared with the GPSR which its performance has been evaluated based on the same scenarios.

5.14.3.1 Network Size

The chart in Figure 5.17 represents the impact of the number of nodes in the network on the time delay involved in delivering the packet to its destination. Overall the chart shows that the end-to-end time delay decreases with increasing network size; this is because more neighbour nodes with an appropriate circumstance become available, which promises a higher level of guarantee that the packets will be forwarded to their destination. Furthermore; we can see from Figure 5.17 that the performance of the GPSR begins to improve initially, then decreases as the number of nodes in the network increases. This is principally due to the fact that the GPSR selects the next-hop node based solely on its position; if no neighbour node with a position closer to the destination than the forwarding node can be found, it will then reach the local maximum (no node is available in the direction of the destination), and will switch to another mode (the perimeter mode), which is not efficient in terms of performance. This leads to the packet being dropped and retransmitted, which increases the time delay. This situation frequently occurs in low density networks. VSHDRP mitigates this drawback by exploiting the SHD and the carry-and-forward approach, which leads from the beginning to avoid sending the packet to a node without an appropriate SHD.

5.14.3.2 Data Packet Size

The chart in figure 5.18 shows the end-to-end time delay against increasing the data packet size; the GPSR shows a slight increase in end-to-end time

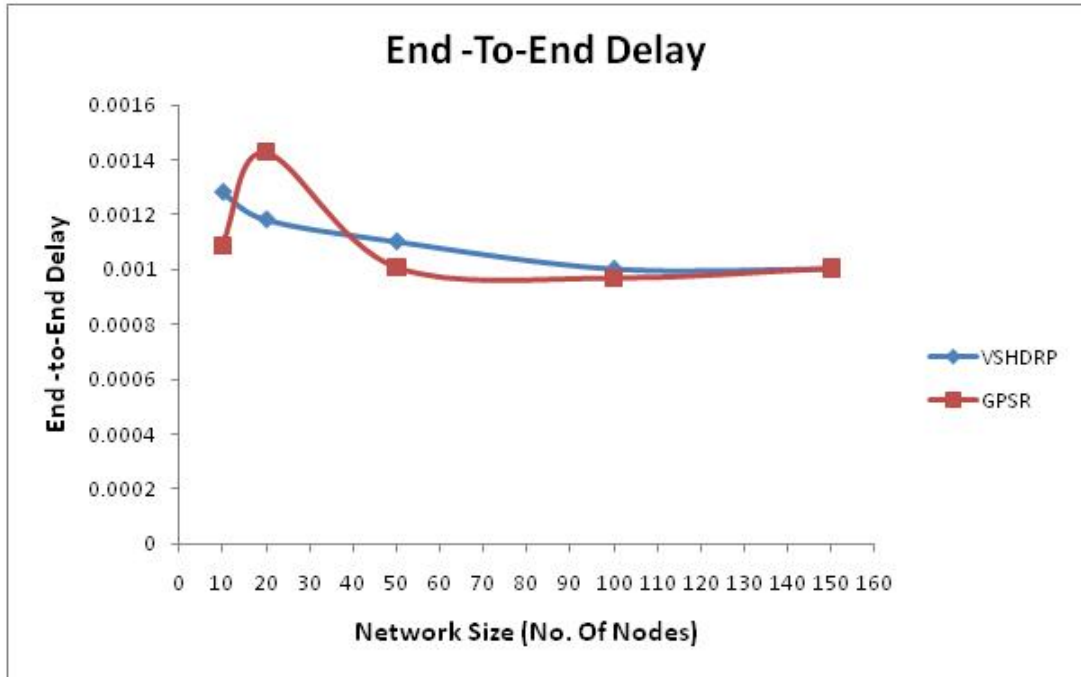


Figure 5.17: Time delay against network size

delay compared with the VSHDRP1 providing less time when increasing the data packet size. This occurs because of the traffic generated in GPSR, which leads to the retransmitting of some packets causing an increase in delay, while in VSHDRP1 the decision about forwarding the packet to the next-hop node was more accurate, taking the other factors into account like the SHD.

5.14.3.3 HELLO beacon Message Period Intervals

The chart in Figure 5.19 shows the end-to-end time delay set against increasing the time intervals when sending the HELLO beacon message. Typically, the VSHDRP shows a slight increase in delay when increasing the period between the sending of the HELLO beacon message.

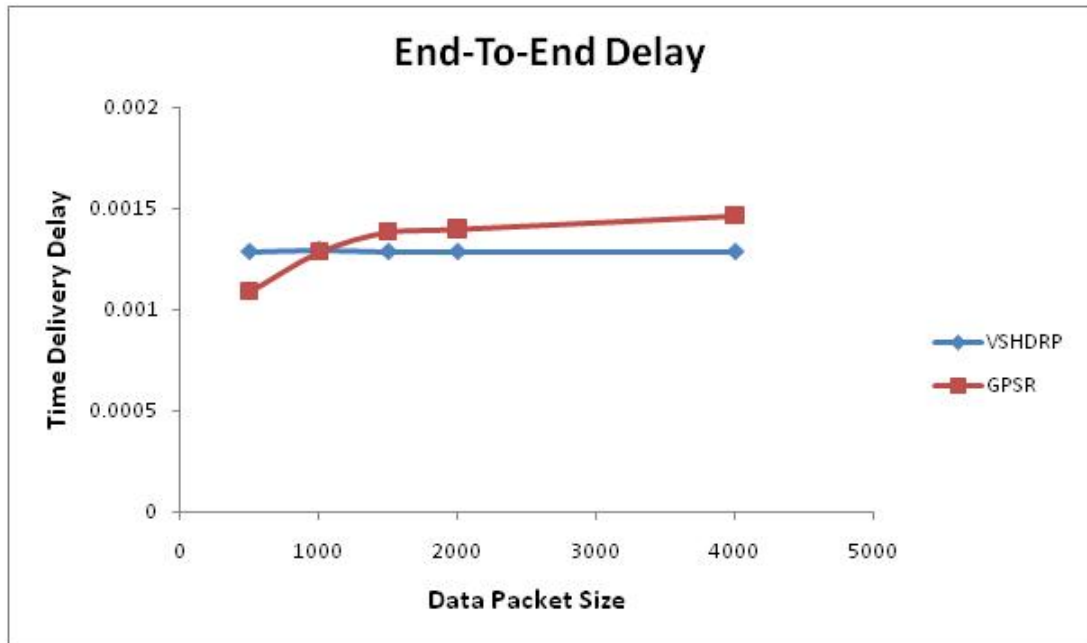


Figure 5.18: Time delay against data packet size

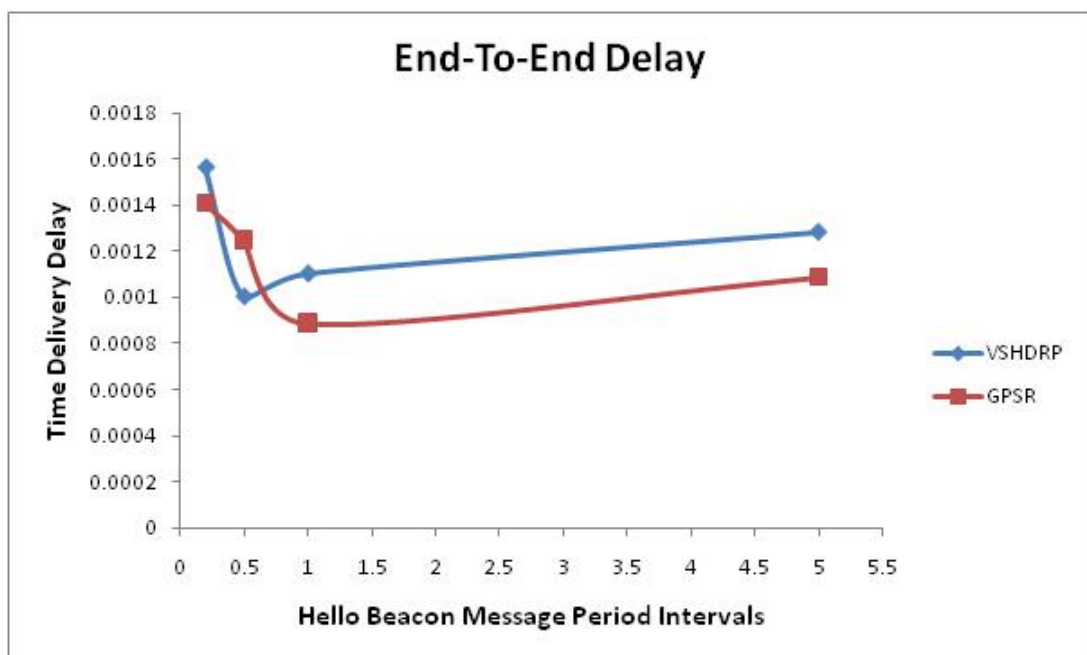


Figure 5.19: Time delay against interval of HELLO beacon message

5.15 Summary

A novel routing protocol mechanism in a highway environment for managing packet delivery issues in VANET; a protocol called VSHDRP1 has been proposed, as a step towards improving the ITS. This protocol assumes that each node can provide some information about its predicted journey; this is considered realistic in regards to the current availability of devices such as GPS and navigation systems.

Exploiting some information from each vehicle can be anticipated to improve the performance of managing the routing protocol in VANET. VSHDRP1 has utilised the knowledge of SHD provided by each vehicle in the network, each vehicle can provide the surrounding neighbouring vehicles with this knowledge by including it in the periodic HELLO beacon message, in conjunction with other information such as position, direction and speed. A filtration process for selecting the next hop-node has been developed, based on four main stages (position, direction, SHD and speed), in order to select an appropriate next-hop node to act as an intermediate node to deliver the packet to its final destination.

The VSHDRP1 has been formalised using the CCA method, in order to prove and validate the routing protocol mechanism. Three different scenarios have been applied, to cover all the algorithm cases. A performance evaluation has been performed, using an NS-2 simulator to show the efficiency of the proposed routing protocol VSHDRP1, comparing its performance with a standard routing protocol (GPSR). The experimental result shows that the VSHDRP1 has

performed well under many tests and in different scenarios.

Chapter 6

Vehicle Second Heading Direction Routing Protocol (VSHDRP2)

Objectives:

- Defining the objectives behind developing the proposed routing protocol VSHDRP2
 - Describing the proposed routing protocol mechanism VSHDRP2 in an Intersection and Roundabout scenario using the aspect of the SHD
 - Implementing the proposed VSHDRP2
 - Presenting the formal specification of VSHDRP2 using CCA
-

6.1 Introduction and Motivation

In this chapter, a novel routing approach for VANET using the multi-hop technique has been presented. In the previous chapter the VSHDRP1 protocol was introduced, representing the standard mode for a highway scenario, where the location of the destination node is considered known. However if the location of the node destination is not precisely known, then the opportunity of delivering the packet to its final destination is low. Therefore VSHDRP1 is not suitable for this case, in other words VSHDRP1 does not take into consideration that the destination node may be located in any direction when an intersection or roundabout is ahead. As a result of this gap, in this chapter VSHDRP2 is proposed to increase the efficiency of the VSHDRP1 to increase the probability of delivering the packet to its final destination; by forwarding the packet in all directions, based on selecting an appropriate node from each direction according to specific criteria.

Proposing a VSHDRP2 is motivated by the need for deploying the aspect of SHD on a roundabout and intersection scenario beside the straight highway scenario in the VSHDRP1, in order to ensure delivering the packet to the destination, especially in the emergency situation as introduced in proposed case study in chapter 7.

VSHDRP1 is more suitable for a highway scenario that does not include intersections and roundabouts; in which vehicles move mostly in the direction of a main road, and when the destination node location is known in which direction. However VSHDRP2 has been proposed as needing to take into con-

sideration the possibilities of facing roundabouts and intersections in a city scenario. Therefore it is insufficient to forward the packet to one node that has not a SHD set on the next exit, it is necessary to forward the packet to at least one node in each direction.

6.2 The proposed Routing Protocol VSHDRP2

In this chapter, the second proposed protocol *Vehicle Second Heading Directional Routing Protocol (VSHDRP2)* has been introduced, the operation of VSHDRP2 is based on the carry-and-forward technique in a DTN network and exploits the knowledge of the SHD of each vehicle to forward the packet to one vehicle from each direction; this section can be considered as containing three subsections:

The first part describes the mobility and nodes classification, the second part introduces the packet sending operation, whilst the third part presents the process of delivery confirmation.

6.2.1 Mobility and Nodes Classification

In VSHDRP2 each node in the network classifies its neighbouring nodes according to their SHD into four different zones direction groups (Z_1 , Z_2 , Z_3 and Z_4), as shown in figure 6.1. The VSHDRP2 protocol demands that each mobile vehicle in VANET sends its mobility information to its neighbouring vehicles periodically. According to the knowledge obtained from the periodic HELLO beacon message, the mobile vehicle in the network is able to group its neighbouring nodes according to their SHD into four zones. Each zone will represent the direction of vehicles as described by a specific angle, for instance

the SHD between 225° and 315° are considered *Zone1*, the SHD between 315° and 45° are considered *Zone2*, the SHD between 45° and 135° are considered as *Zone3* and the SHD between 135° and 225° are considered as *Zone4*. In other words if the direction of the next road that needs to be taken by a vehicle is 270° that means the SHD of this vehicle falls under *Zone1* and so on.

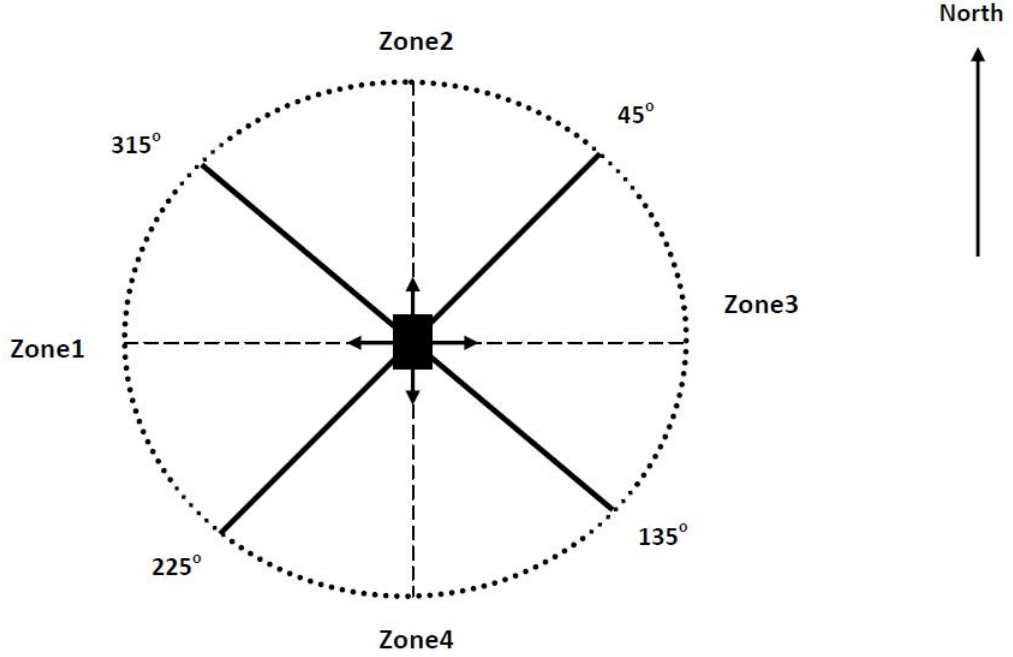


Figure 6.1: The classification of neighbour nodes according to the SHD angles

If a source node wants to send a packet to a destination node, the source node acquires the knowledge of its own location, current direction, velocity and SHD through GPS, NS and preloaded digital road maps; in addition to this, the source node acquires sufficient knowledge relating to position, velocity, current direction, SHD and the zone number of neighbouring nodes from periodic beacon messages, are shown below in table 6.1.

Field Name	Description
Node-IP	Node IP Address
Node-dir1	Node First Direction (Current road direction)
Node-dir2	Node Second Heading Direction SHD (next road)
Node-pos	Node Current Position
Node-vlo	Node Moving Velocity
Nb	Information about its neighbour nodes
Node- Zone	A zone of Second Heading Direction Number(1-4)

Table 6.1: HELLO beacon message format for VSHDRP2

6.2.2 Packet Sending Process

As in VSHDRP1, VSHDRP2 works under the assumption that the type of the network is DTN (a disconnected area can be generated any time when no vehicles are available in a specific area) and each node in the network has sufficient knowledge about its position and directions and predetermined route for its journey, as utilised by the GPS and NS when a source node has a packet that needs to be sent. As illustrated in the flowchart in figure 6.2, when a source node S needs to send a packet to a destination node D , it will look for the destination node in its cache (neighbours table), and if the destination node is found to be a neighbour in its cache, S will start forwarding the data packets to node D . If D is not found in the cache of the source node S , then node S will set the current direction and the zones according to the SHD, then start to look for an appropriate next-hop node using the filtration process for selecting the next-hop node.

The process of selecting an appropriate next-hop node needs to follow these four stages in view of the following:

- (a) According to their position
- (b) According to their current direction

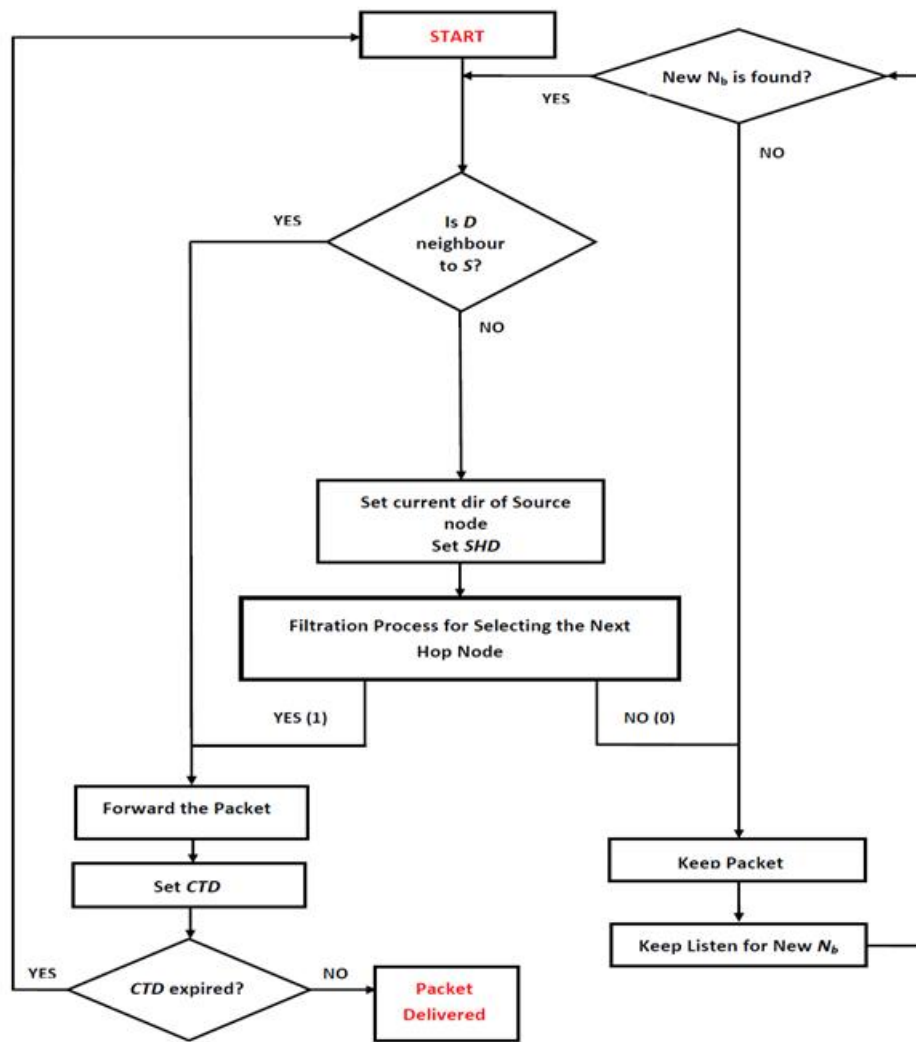


Figure 6.2: VSHDRP2 algorithm

(c) According to their second direction

(d) According to their speed

- As shown in flowchart in figure 6.3, if a neighbour node having the desired position (closer to the destination than node S) is found, then S will check for the current direction of these neighbours:
 - If a neighbour node travelling in the same current direction (current road) as the S node is found (current direction = 0), then the S node will check if there is a roundabout ahead or an intersection by

exploiting the knowledge of the digital road map, if one of these is found ahead, then it checks the SHD (next road):

- * If a neighbour with an appropriate SHD is found by the S node in its cache, the S node forwards the packet to this node. If more than one node has an appropriate SHD, then the S node will select a node from each of the four zones (that we are classified in the beginning of the chapter). In instances where more than one appropriate node is available in each zone, the node with the highest speed will be chosen.
- * If no neighbour node with an appropriate SHD can be found in its cache, node S will look for neighbours of neighbours that have an appropriate SHD. If more than one neighbour is found, the node with the highest speed will be chosen, otherwise it will retain the packet.
- If no neighbour node with the same current direction is found (opposite direction is considered), which means current direction = 1, then node S will look for nodes that have the same current direction in its neighbours of neighbours table:
 - * If an appropriate node is found, then S will check for its SHD, if more than one appropriate node is found it will select one from each zone.
 - * If no neighbour can be found, there will be a disconnected area separating the source node from the other nodes in the network. Thus, the source neighbour will buffer the packet until a new node appears in its cache of neighbours.
- If there are no neighbour nodes with an appropriate position, then the S

node will keep holding the packet until a new node is available.

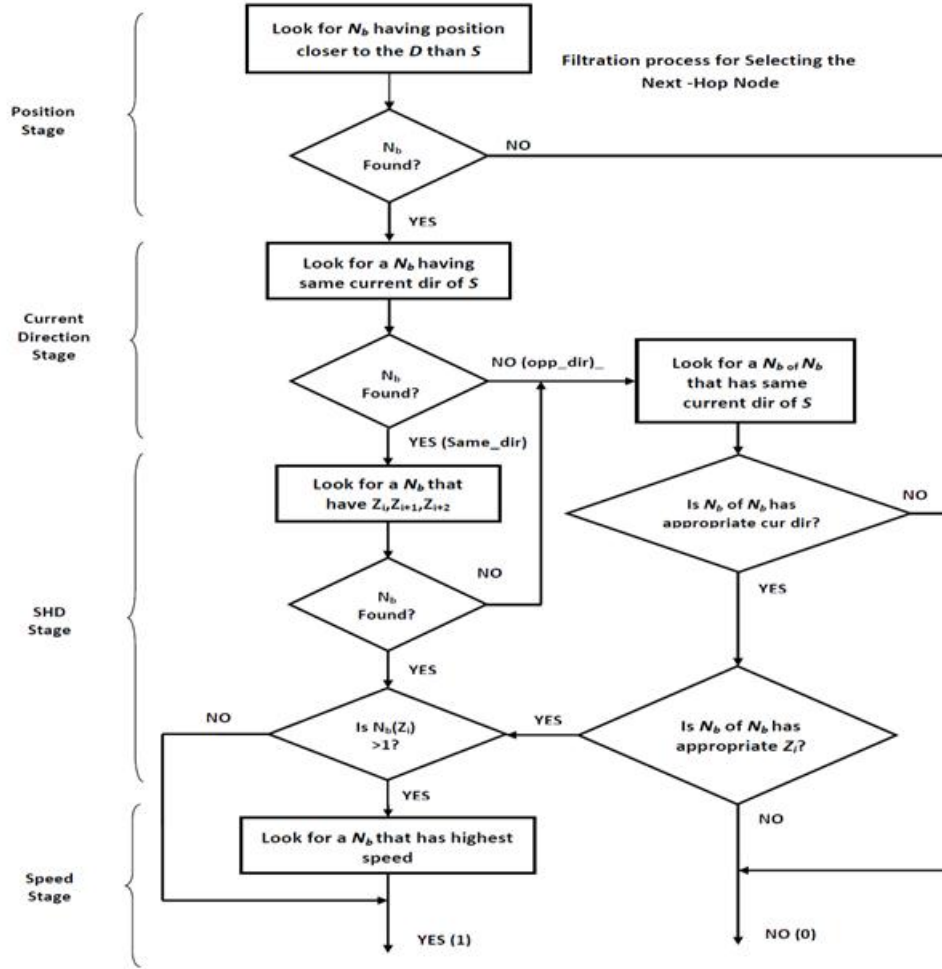


Figure 6.3: Filtration process for selecting the next-hop node

6.2.3 Packet Delivery Confirmation

As in VSHDRP1, when the forwarder node intends to forward the packet to another intermediate node, it must ensure that the packet has been delivered successfully to that node, therefore it is important at this stage to send back an acknowledgement for the sending process representing by the CTD packet.

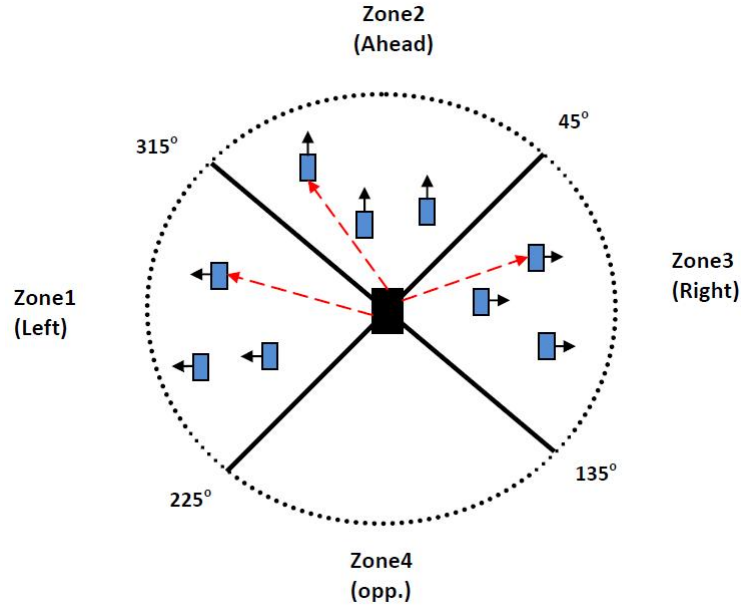


Figure 6.4: The node distribution based on the SHD

6.3 Example of packet propagation process in VSHDRP2

As shown in figure 6.4 the node intending to forward the packet to the node destination is willing to classify its neighbour nodes according to its SHD. In this section the algorithm of VSHDRP2 has been illustrated in the following example; as shown in figure 6.5, the node S is intending to send a packet in all directions after detecting an intersection ahead; therefore node S will send the packet to one node from each direction to guarantee the packet is delivered to the final destination (RSU) in each direction, if we assume that this packet is an emergency packet to alert other vehicle about an accident occurrence. In figure 6.5 the node S is willing to select the next hop-node according to the position of each vehicle (the closest vehicle to the destination than node S), nodes I_1 , I_2 and I_3 have been selected as the appropriate nodes to hold the packet towards the destinations.

Figure 6.6 shows that node S will forward the packet to the left through node I_1 and ahead through I_2 , while for the right option it will exploit the node in the opposite direction I_4 because it has a neighbour I_3 willing to take it in right direction. In case there are no nodes available in the current direction, then the node that holds the packet will check if the nodes in the opposite direction have a suitable neighbouring node.

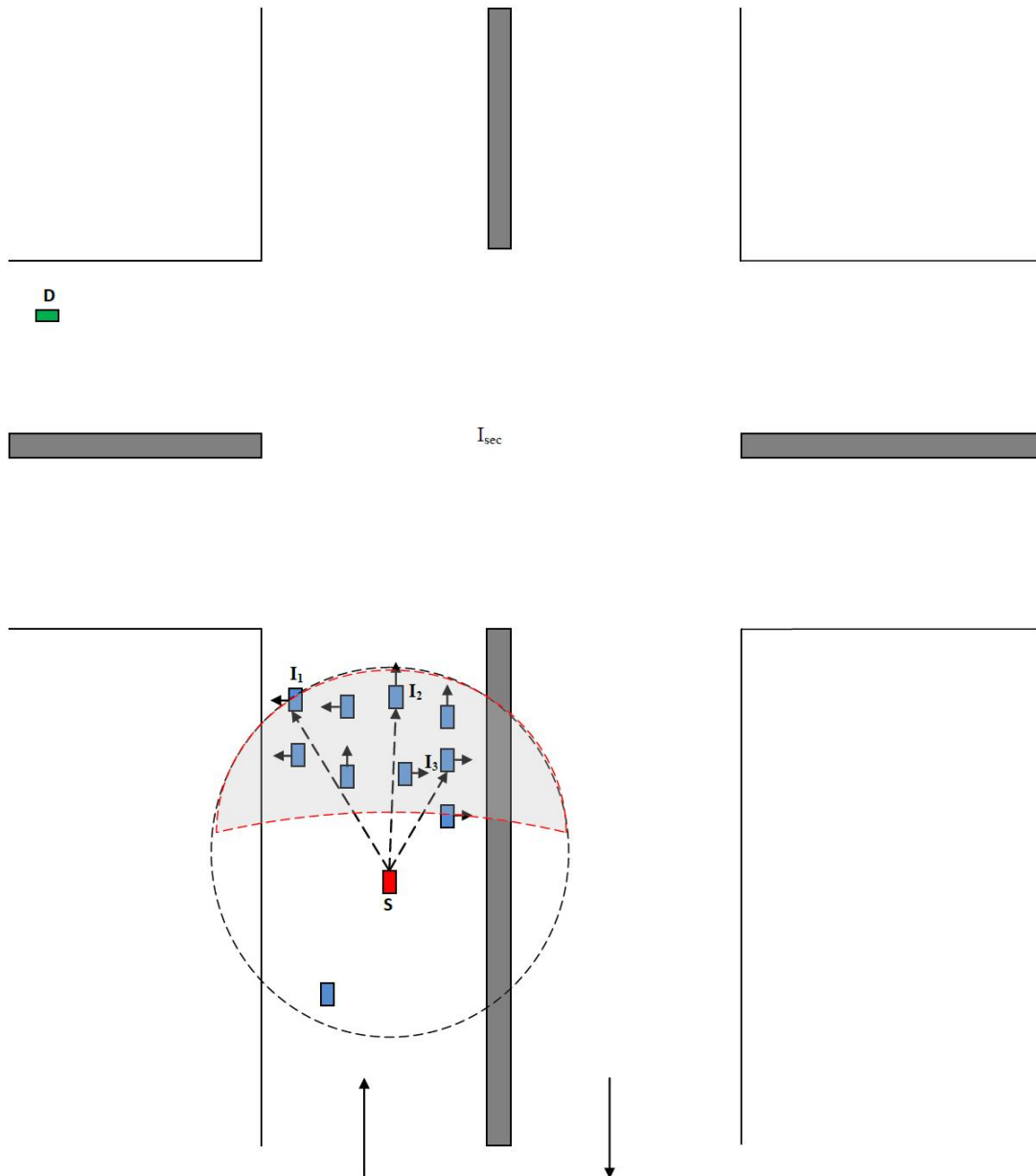


Figure 6.5: Next hop-node selection based on position

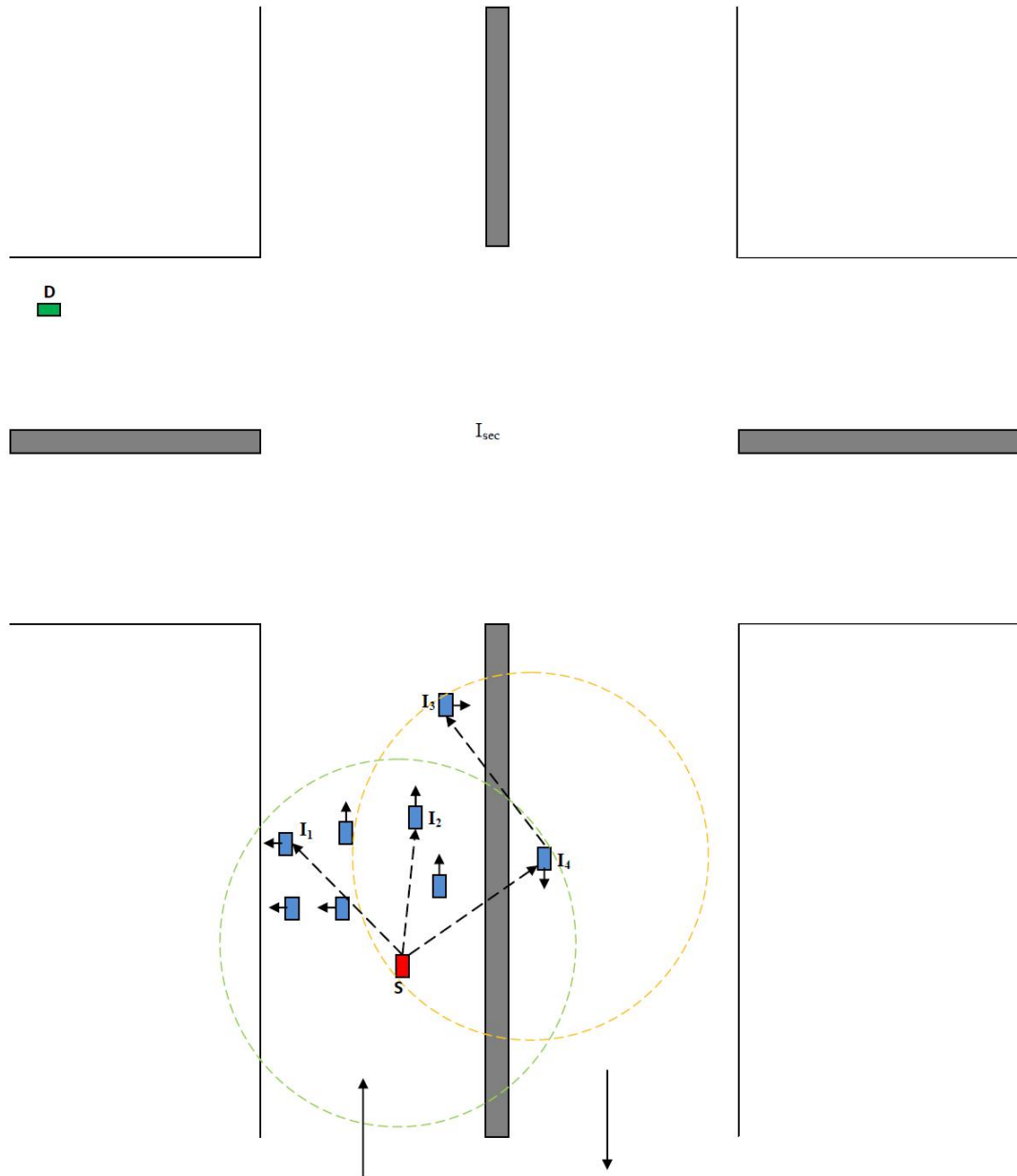


Figure 6.6: Next hop-node selection based on SHD

6.4 Formal Specification of VSHDRP2

We now give a formal specification of the VSHDRP2 protocol in CCA.

6.4.1 System Model

A VANET is modelled in CCA as a parallel composition of all the nodes in the network (e.g. vehicles and road side units), i.e.

$$VANET \hat{=} node_0 \mid node_1 \mid \dots \mid node_{k-1} \quad (6.1)$$

Each node, $node_i$, in the VANET is modelled as an ambient of the following structure:

$$id[P_{id} \mid NH1[P_1] \mid NH2[P_2] \mid Hspeed_j[P_3] \mid Lspeed_j[P_4] \mid SHD_j[P_5]] \quad (6.2)$$

for $j = 1, 2, 3, 4$

where

- The index $j = 1, 2, 3, 4$ is the zone number.
- id is the node's id. For the sake of simplicity, we use SN to denote the source node, DN for the destination node and IN_m for intermediate nodes, $m \geq 0$.
- P_{id} is a process that specifies the capabilities of the node, e.g. its ability to communicate or to sense the presence of other nodes in its range.
- $NH1$ is an ambient that contains the ids of the neighbouring node moving in the direction of the destination node (aka direction 1) and closer to the destination node.
- $NH2$ is an ambient that contains the ids of the neighbouring node moving in the direction opposite to direction 1 (this is called direction 2) and closer to the destination node.

- $Hspeed_j$ is an ambient that contains the ids of the neighbouring nodes moving in high speed (i.e. speed greater than or equal to a specified threshold) with second heading direction (shd) in zone j , $j = 1, 2, 3, 4$.
- $Lspeed_j$ is an ambient that contains the ids of the neighbouring nodes moving in low speed (i.e. speed less than the threshold) with shd in zone j , $j = 1, 2, 3, 4$.
- SHD_j is an ambient that contains the ids of the neighbouring nodes with shd in zone j , $j = 1, 2, 3, 4$.
- Each process P_m , $1 \leq m \leq 5$, is either the inactivity process $\mathbf{0}$ or a parallel composition of ambients of the form $n[\mathbf{0}]$ where n is a node's id.

Each run of the VSHDRP2 protocol may involve up to three types of nodes: one source node, zero or many intermediate nodes and one destination node. The source node is the node that initiates the run and the destination node is the node the message is intended to. When the destination node is not in range with the source node, intermediate nodes are used to build a route to the destination node. We now specify the behaviours proper to each type of node (source node, intermediate node and destination node). Of course, each node in a VANET should be able to act as a source node, an intermediate and a destination node.

6.4.2 Context Expression

The context expressions used in the specification of the VSHDRP2 protocol are summarised as follows, where $j = 1, 2, 3, 4$ is the zone number:

- $hasNb1(n)$ holds if the node n is a neighbour closer to destination in direction 1, i.e. $hasNb1(n) \triangleq \spadesuit(\bullet \mid NH1[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$.

- $hasNb2(n)$ holds if the node n is a neighbour closer to destination in direction 2, i.e. $hasNb2(n) \hat{=} \spadesuit(\bullet \mid NH2[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$.
- $hasNb(n)$ holds if the node n is a neighbour closer to destination (regardless its direction), i.e. $hasNb(n) \hat{=} hasNb1(n) \vee hasNb2(n)$.
- $noNb1()$ holds if there are no neighbouring nodes in direction 1, i.e. $noNb1() \hat{=} \neg\spadesuit(\bullet \mid NH1[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True})$.
- $hasSHD_j(n)$ holds if the node n is a neighbour with second heading direction in zone j , i.e. $hasSHD_j(n) \hat{=} \spadesuit(\bullet \mid SHD_j[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True})$.
- $noSHD_j()$ holds if there are no neighbours with SHD in zone j , i.e. $noSHD_j() \hat{=} \neg\spadesuit(\bullet \mid SHD_j[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True})$.
- $hasNbofNb_j(n)$ holds if node n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1, i.e. $hasNbofNb_j(n) \hat{=} hasNb2(n) \wedge \spadesuit(n[\mathbf{True} \mid SHD_j[\mathbf{True} \mid \oplus \mathbf{True}]] \mid \mathbf{True})$.
- $highestSpeed_j(n)$ holds if the node n has the highest speed and a shd in zone j , i.e. $highestSpeed_j(n) \hat{=} \spadesuit(\bullet \mid Hspeed_j[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}) \vee (\neg\spadesuit(\bullet \mid Hspeed_j[\mathbf{True} \mid \oplus \mathbf{True}] \mid \mathbf{True}) \wedge \spadesuit(\bullet \mid Lspeed_j[\mathbf{True} \mid n[\mathbf{True}]] \mid \mathbf{True}))$

6.4.3 Source Node

A source node SN is the node that initiates a run of the VSHDRP2 protocol, willing to send a message msg to a destination node DN . So its capabilities

are modelled by the following process:

$$P_{SN} \hat{=} (\nu z1) (\nu z2) (\nu z3) (\nu z4) \left\{ \begin{array}{l} Eq.6.4 \mid Eq.6.5 \mid Eq.6.6 \mid \\ z1[\uparrow send().\mathbf{0}] \mid z2[\uparrow send().\mathbf{0}] \mid \\ z3[\uparrow send().\mathbf{0}] \mid z4[\uparrow send().\mathbf{0}] \end{array} \right\} \quad (6.3)$$

Where the restricted names $z1$, $z2$, $z3$ and $z4$ are used to guarantee that not more than one of the processes specified in Eq.6.4, Eq.6.5 and Eq.6.6 are executed, for each zone. These processes are specified as follows:

- if the destination node is a neighbour, send message to destination node and wait for acknowledgement. This is formalised as:

$$hasNb(DN)?z1\downarrow\mathbf{recv}().DN :: \mathbf{send}(SN, msg).DN :: \mathbf{recv}(y).\mathbf{0} \quad (6.4)$$

- if the destination node is not a neighbour then for each zone j , look for an intermediate node moving in direction 1 with shd in zone j and highest speed, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \mathbf{find} \ n : E_1^1(n, DN) \ \mathbf{for} \ z1\downarrow\mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).\mathbf{0} \mid \\ \mathbf{find} \ n : E_2^1(n, DN) \ \mathbf{for} \ z2\downarrow\mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).\mathbf{0} \mid \\ \mathbf{find} \ n : E_3^1(n, DN) \ \mathbf{for} \ z3\downarrow\mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).\mathbf{0} \mid \\ \mathbf{find} \ n : E_4^1(n, DN) \ \mathbf{for} \ z4\downarrow\mathbf{recv}().n :: \mathbf{send}(SN, DN, msg).n :: \mathbf{recv}(y).\mathbf{0} \end{array} \right\} \quad (6.5)$$

where $E_j^1(s, t) \hat{=} \neg hasNb(t) \wedge hasNb1(s) \wedge hasSHD_j(s) \wedge highestSpeed_j(s)$, for $j = 1, 2, 3, 4$.

- if no such intermediate nodes exist for a zone j then look for an intermediate node moving in direction 2 which has a neighbour with shd in zone j moving in direction 1 and closer to the destination node, and send the

message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \text{find } n : E_1^2(n, DN) \text{ for } z1 \downarrow \text{recv}().n :: \text{send}(SN, DN, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_2^2(n, DN) \text{ for } z2 \downarrow \text{recv}().n :: \text{send}(SN, DN, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_3^2(n, DN) \text{ for } z3 \downarrow \text{recv}().n :: \text{send}(SN, DN, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_4^2(n, DN) \text{ for } z4 \downarrow \text{recv}().n :: \text{send}(SN, DN, msg).n :: \text{recv}(y).0 \end{array} \right\} \quad (6.6)$$

where $E_j^2(s, t) \hat{=} \neg hasNb(t) \wedge noSHD_j() \wedge hasNbofNb_j(s)$

6.4.4 Intermediate Node

An intermediate node *IN* receives a triple (*sender*, *dest*, *msg*) where *sender* is the sender's id, *dest* is the destination node's id and *msg* is the message being sent. The intermediate node confirms the receipt by sending an acknowledgement to the sender and forwards the message to an appropriate node in each zone. This is specified as:

$$\begin{aligned} P_{IN} \hat{=} & ! :: \text{recv}(sender, dest, msg).sender :: \text{send}(ack).(\nu z1) (\nu z2) (\nu z3) (\nu z4) \{ \\ & Eq.6.8 \mid Eq.6.9 \mid Eq.6.10 \mid z1[\uparrow send().0] \mid z2[\uparrow send().0] \mid \\ & z3[\uparrow send().0] \mid z4[\uparrow send().0] \} \end{aligned} \quad (6.7)$$

where the restricted name *z1*, *z2*, *z3* and *z4* play the same role as in Eq.6.3 for selecting at most one of the processes Eq.6.7, Eq.6.8 and Eq.6.9, for each zone. The replication operator '!' means that an intermediate node repeats this pattern of behaviour its whole lifetime. Moreover, an intermediate node determines the next node to forward the message to as follows:

- if the destination node *dest* is a neighbour, send message to destination node and wait for acknowledgement, i.e.

$$hasNb(dest)?z1 \downarrow \text{recv}().dest :: \text{send}(IN, msg).dest :: \text{recv}(y).0 \quad (6.8)$$

- if the destination node $dest$ is not a neighbour then for each zone j , look for another intermediate node moving in direction 1 with SHD in zone j and highest speed, and send the message to that intermediate and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \text{find } n : E_1^1(n, dest) \text{ for } z1 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_2^1(n, dest) \text{ for } z2 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_3^1(n, dest) \text{ for } z3 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_4^1(n, dest) \text{ for } z4 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \end{array} \right\} \quad (6.9)$$

- if no such intermediate nodes exist for a zone j then look for an intermediate node moving in direction 2 which has a neighbour with shd in zone j moving in direction 1 and closer to the destination node, and send the message to that intermediate node and wait for acknowledgement, viz.

$$\left\{ \begin{array}{l} \text{find } n : E_1^2(n, dest) \text{ for } z1 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_2^2(n, dest) \text{ for } z2 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_3^2(n, dest) \text{ for } z3 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \mid \\ \text{find } n : E_4^2(n, dest) \text{ for } z4 \downarrow \text{recv}().n :: \text{send}(IN, dest, msg).n :: \text{recv}(y).0 \end{array} \right\} \quad (6.10)$$

6.4.5 Destination Node

The destination node DN receives a pair $(sender, msg)$ and sends an acknowledgement to the sender. This behaviour is formalised as:

$$P_{DN} \hat{=} ! :: \text{recv}(sender, msg).sender :: \text{send}(ack).0$$

6.4.6 Analysis of the Protocol

The formal specification of the VSHDRP protocol presented above is executable by the CCA interpreter. In this section, we run the protocol in a

variety of scenarios reflecting differing configurations of the VANET network. In these scenarios, a source node SN is willing to send a packet to a destination node DN . Other nodes of the network are named $IN1$, $IN2$ and so on. The result of each run of the protocol is a sequence of reductions showing the interactions that happened among the VANET's nodes. This output can then be analysed to detect flaws in the protocol at an early stage, prior to implementation and deployment. The full formal executable specification of VSHDRP2 is illustrated in Appendix C. For illustration, we consider three scenarios as follows.

6.4.6.1 Scenario 1

In this scenario, the destination node is not neighbour to the source node; rather the following neighbourhood relationship is considered: $\{(SN, IN1), (SN, IN2), (SN, IN3), (IN3, DN)\}$. In addition, the nodes SN , $IN1$, $IN2$ and $IN3$ move in direction 1; $IN1$ has shd in zone 1, $IN2$ in zone 2 and $IN3$ in zone 3. The output of the protocol's run is given in Table 6.2.

```

1. ---> {binding: n -> IN1}
2. ---> {binding: n -> IN2}
3. ---> {binding: n -> IN3}
4. ---> {Child to parent: x$0 ===()===> SN}
5. ---> {Child to parent: y$1 ===()===> SN}
6. ---> {Child to parent: z$2 ===()===> SN}
7. ---> {Sibling to sibling: SN ===(SN,DN,packet)===> IN1}
8. ---> {Sibling to sibling: IN1 ===(ack)===> SN}
9. ---> {Sibling to sibling: SN ===(SN,DN,packet)===> IN2}
10.---> {Sibling to sibling: IN2 ===(ack)===> SN}
11.---> {Sibling to sibling: SN ===(SN,DN,packet)===> IN3}
12.---> {Sibling to sibling: IN3 ===(ack)===> SN}
13.---> {Child to parent: x$220 ===()===> IN3}
14.---> {Child to parent: y$221 ===()===> IN3}
15.---> {Child to parent: z$222 ===()===> IN3}
16.---> {Sibling to sibling: IN3 ===(IN3,packet)===> DN}
17.---> {Sibling to sibling: DN ===(ack)===> IN3}

```

Table 6.2: Output of scenario 1

In line 1, 2 and 3 in table 6.2 the notation {binding: n -> IN1, IN2 and IN3} corresponds to the execution of a *search prefix* and means that the node IN1, IN2 and IN3 are found (by the source node SN) as an appropriate nodes to forward the message to. In line 4, 5 and 6, the source node SN detects that the Intermediate node IN1 is a neighbour in zone1, IN2 is a neighbour in zone2 and IN3 is a neighbour in zone3 and decides to execute the process in Eq.6.4. Then the packet is sent directly to those nodes as shown in lin7, line9 and line11 repesctively, which acknowledges in line 8, 10 and 12 respectively In line 13, the IN3 detects that the destination node DN is a neighbour, after that IN3 send the packet to DN as shown in line 16, which acknowledged in line 17 and the run terminates.

6.4.6.2 Scenario 2

In this scenario, the destination node is not neighbour to the source node; rather the following neighborhood relationship is considered: $\{(SN, IN1), (SN, IN2), (SN, IN4), (IN4, IN3), (IN3, DN)\}$. In addition, the nodes SN, IN1, IN2 and IN3 move in direction 1, while IN4 moves in direction 2; IN1 has shd in zone 1, IN2 in zone 2 and IN3 in zone 3. The output of the protocol's run is given in Table 6.3.

1. --->	{binding: n -> IN1}
2. --->	{binding: n -> IN2}
3. --->	{binding: n -> IN4}
4. --->	{Child to parent: x\$0 ===()===> SN}
5. --->	{Child to parent: y\$1 ===()===> SN}
6. --->	{Child to parent: z\$2 ===()===> SN}
7. --->	{Sibling to sibling: SN ===(SN,DN,packet)===> IN1}
8. --->	{Sibling to sibling: IN1 ===(ack)===> SN}
9. --->	{Sibling to sibling: SN ===(SN,DN,packet)===> IN2}
10. --->	{Sibling to sibling: IN2 ===(ack)===> SN}
11. --->	{Sibling to sibling: SN ===(SN,DN,hello)===> IN4}
12. --->	{Sibling to sibling: IN4 ===(ack)===> SN}
13. --->	{binding: n -> IN3}
14. --->	{Child to parent: z\$196 ===()===> IN4}
15. --->	{Sibling to sibling: IN4 ===(IN4,DN,hello)===> IN3}
16. --->	{Sibling to sibling: IN3 ===(ack)===> IN4}
17. --->	{Child to parent: x\$175 ===()===> IN3}
18. --->	{Child to parent: y\$176 ===()===> IN3}
19. --->	{Child to parent: z\$177 ===()===> IN3}
20. --->	{Sibling to sibling: IN3 ===(IN3,hello)===> DN}
21. --->	{Sibling to sibling: DN ===(ack)===> IN3}

Table 6.3: Output of scenario 2

In line 1, 2 and 3 in table 6.3 the notation {binding: n -> IN1, IN2 and IN4} corresponds to the execution of a *search prefix* and means that the node IN1, IN2 and IN4 are found (by the source node SN) as an appropriate nodes to for-

ward the message to. However node IN3 is direction 2 (opposite direction),but IN3 have a neighbour node that belong to zone3, which will recieve the packet and forward it to the destination node DN as shown in line 20.

6.4.6.3 Scenario 3

In this scenario, the destination node DN is a neighbour to the source node SN; therefore the packet will be sent directly to the DN without any intermediate node, although there are three intermediate nodes are available as a neighbour to SN

```

1.---> {Child to parent: x$0 ===()===> SN}
2.---> {Child to parent: y$1 ===()===> SN}
3.---> {Child to parent: z$2 ===()===> SN}
4.---> {Sibling to sibling: SN ==(SN,packet)==> DN}
5.---> {Sibling to sibling: DN ==(ack)==> SN}
    
```

Table 6.4: Output of scenario 3

In line 1,2 and 3 in table 6.4 the source node SN detects that some nodes are neighbours including the destination node DN. In line 4 the source node SN sending the packet to the destination node DN, then it acknowledges in line 5

6.5 Roundabout/Intersection Scenario

In this scenario the vehicles may have many roads options to take where there are several directions for vehicles such as at the roundabout or intersection where there is more than one exist as shown in figure 6.7. And if the forwarding vehicle need to send a message to the nearest ambulance vehicle, which there is no information available about its exact location, here the forwarding node

need to guarantee delivering the message to this ambulance vehicle, in order to request it help for an accident occurrence

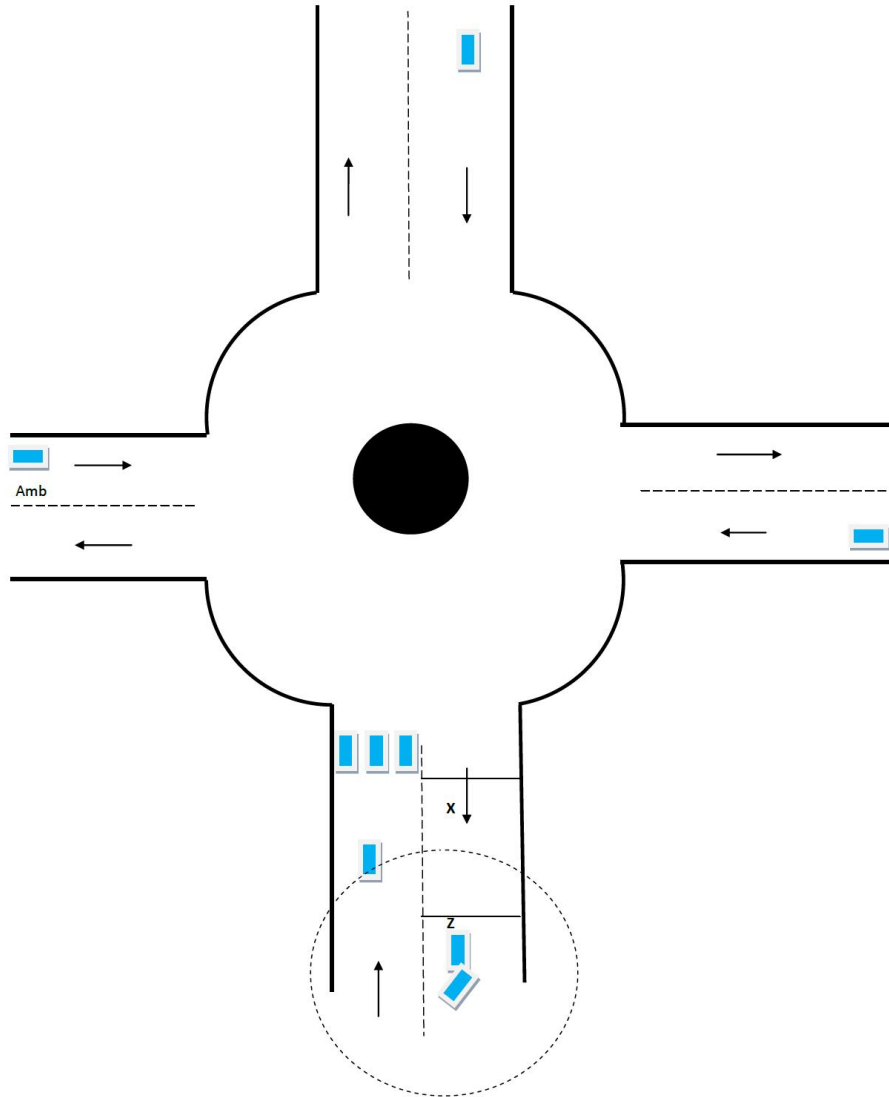


Figure 6.7: Roundabout Scenario

Hence, in this situation the VSHDRP2 is considered as the proper routing protocol to be applied, as it working on sending the message in all directions by selecting one node in each direction to forward the message in that direction. Assume an accident occurs in a road in a sparse environment, as shown in Figure 6.4; node Z sends a warning message that an accident has occurred in the specific location, in order to request an ambulance vehicle Amb as

well as to advise other vehicles that are approaching this location to avoid the jam; at the same time, there is no vehicle available at section X of the road to hold the warning message to alert the other incoming vehicles from other side, as shown in the figure 6.7. There are no vehicles available at the roundabout; therefore the carry-and-forward strategy is need to be active in this situation, by exploiting the vehicles available at the opposite side. Vehicle holding warning message can select one vehicle from the second road direction (going north) group, as shown in figure 6.4, to ensure delivery of the message to all vehicles when approaching the roundabout.

6.6 NS-2 Simulation

Because a simulation has been done for the first mode VSHDRP1, and the concept of using the SHD has shown a better performance than the other routing protocol GPSR in terms of delivery ratio, overhead and time delay, we found there is no purpose from making a simulation for the second mode VSHDRP2.

6.7 Summary

In this chapter a novel routing protocol mechanism in VANET has been proposed, called VSHDRP2 as the second mode for the VSHDRP; this mode has been developed for the intersection and roundabout environment to manage packet delivery issues when the location of the destination node is not exactly known, as a step toward improving the ITS. This protocol assumes that each node can provide some information about its predicted journey, which is considered realistic in view of the availability of devices such as GPS and

navigation systems.

VSHDRP2 has utilised knowledge of SHD provided by each vehicle in the network. Each vehicle can provide neighbouring vehicles with this knowledge by including them in the periodic HELLO beacon message, in addition to offering other information such as position, direction and speed.

A filtration process for selecting the next hop-node has been developed, based on four main stages (position, direction, SHD and speed), in order to select an appropriate next-hop node to act as an intermediate node to deliver the packet to its final destination.

The VSHDRP2 has been formalised using the CCA method, in order to prove and validate the routing protocol mechanism. Three different scenarios have been applied, to cover all the cases described by the algorithm.

Chapter 7

Conclusion and Future Work

Objectives:

- Summarise the work in this thesis
 - List the main contributions of this work
 - Propose future work that follows on from this thesis
-

7.1 Conclusion

VANET is considered to be a special case of MANET; in which vehicles acts as nodes in a network, having a vital impact on wireless communications on the roads, in order to accomplish many applications that need to be applied using the VANET to promise safety and comfort. The increasing numbers of fatal accidents on the roads, which are caused by vehicle collisions and traffic congestion, have focused the attentions of governments, car manufacturers and researchers on improving the field of Intelligent Transport systems, in an attempt to reduce the percentage of fatal accidents occurring on the roads. Indeed VANET has played an essential and vital role in supporting intelligent transport systems.

The work that has conducted in this thesis is focused to find a robust route packet to the desired destination, in order to increase the connectivity and satiability, which lead then to increase the network reliability, in terms of increasing delivery ratio, reducing overhead and time delay.

Solving this problem will lead to achieve the main aim of increasing the efficiency of safety applications in vehicular ad hoc network, and that is accomplished by improving the routing management in VANET. This aim is achieved by contributions introduced in this thesis.

The first contribution is presented in chapter 4, which introduced a new technique for organising the broadcasting of HELLO beacon message in VANET by making it adaptively rather than periodically, based on the circumstances

of the vehicle such as speed and direction. Any change at least in one of these factors will lead to a change in the network topology, and then the HELLO beacon message is needed to be sent, to inform other vehicles in the networks about its new situation. As a result this technique is worked on eliminating the unnecessary and redundant messages, which led to a reduction in the generated overheads.

Moreover, a novel OBU architecture is presented in this chapter, which is built based on the technique of the adaptive HELLO beacon message, this architecture is utilised the concept of the context-aware system, by dividing the architecture in three main phases; knowledge collective phase, processing phase and reacting phase. This is considered as the first context-aware protocol for VANET.

The second contribution is introduced in chapter 5 and chapter 6, where a new routing protocol is developed for sparse environment in VANET, where the network density is considered to be low, which create a region of gaps between vehicles in the network, that means no vehicles is available to act as intermediate node, which preventing sent packets to be delivered to their destinations.

The proposed protocol is utilised a new parameter called the SHD, which represents the second road the vehicle is intending to follow during its journey. A filtration process is utilised in this protocol to select the next hop node, which comprises four stages; vehicle position, current direction, SHD and Speed.

This is the first thesis that introduced the using of formalisation method to validate a routing protocol in VANET. The behaviour of protocol is validated using the CCA as a formalisation method, the result of the executable specification of the proposed routing protocol VSHDRP show the flaw of the packet from the source to the destination in three different scenarios, which work as a verification to the protocol mechanism. The formalisation is applied on the two modes of the proposed protocol, the highway mode and the roundabout mode.

The simulation result of VSHDRP has shown a better performance over the GPSR, in terms of delivery ratio, overhead and time delay, based on varying three different parameters; network size, data packet size and intervals of broadcasting HELLO beacon messages.

7.2 Achieved Aims

The research problems illustrated in chapter 1 were achieved as follows:

- By using the NS-2 simulator, the performance of the proposed routing protocol VSHDRP has show a better performance over GPSR, in terms of delivery ratio, overhead and time delay, that led to increase route link stability.
- The SHD routing solution achieved the target of increasing the network reliability and connectivity by using a new aspects of the SHD for the process of selecting the next-hop node.
- A novel formal specification of the VSHDRP is performed using the Calculus of Context-aware Ambients (CCA), for the sake of behaviour valida-

tion and verification. In addition the executable code of this specification has run in three different scenarios in each mode, and the result has shown the transactions of packet and how it delivered from the source node to the destination node.

- The target of governing the redundant and unnecessary beacon message was achieved by introducing a new technique, which involved organising the broadcasting process of the HELLO beacon message depending on the circumstances (speed and direction) of each vehicle in the network.
- A novel architecture has introduced, which designed based on the concept of a context-aware system and built for sending the HELLO beacon message adaptively in VANET rather than periodically.

7.3 Future work

The vehicular ad hoc network remains a hot research topic in the field of wireless communications and networking attracting a considerable number of researchers. Many sectors in this field still need to be resolved; at the present time the focus is on proposing suitable solutions to mitigate the problems that prevent progress in this field.

Accordingly, future concerns for this area of research that should be taken into consideration are as follows: In accordance with the technique of adaptive HELLO messages in VANET, an inward examination into the selection parameters that reflect the vehicle's circumstances (In addition to speed and direction) is required, for the sake of improving performance. In addition a formal specification is suggested alongside the proposed architecture, to pro-

vide greater confirmation in terms of improving the system's performance.

More investigation is needed in terms of security as using the second heading direction is one of the factors that is utilised to select the next-hop node, a security platform also needs to be applied to ensure no exploitation of the second heading direction information to intrude on or influence other vehicles' movements.

Implementation of the second mode when using the proposed routing protocol in NS-2 is applied to show the performance of the protocol in terms of delivery ratio, overheads and time delay.

Many other challenges relate to considerable issues that need to be mitigated:

- Cross-layer design for VANET;
- Determining priority sending/receiving and scheduling process in VANET;
- Bay parking applications in VANET
- Detecting and analysing drivers' behaviour to resolve safety issues;
- Security and privacy management in VANET;
- MAC layer protocol development;
- End-to-end quality of service (QoS) provision.

Bibliography

- [1] S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: The incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9:50–61, 2005.
- [2] A.H. Al-Bayatti. Security management for mobile ad hoc network of networks (manon. 2009.
- [3] Car 2 Car Communication Consortium. Mission and objectives. <http://www.car-to-car.org/index.php?id=31> Retrieved on 17.02.2010.
- [4] M. Saito, J. Tsukamoto, T. Umedu, and T. Higashino. Evaluation of Inter-Vehicle Ad-hoc Communication Protocol. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pages 78–83, 2005.
- [5] M. Zhang and R. S. Wolff. Routing Protocols for Vehicular Ad Hoc Networks in Rural Areas. *IEEE Communication Magazine*, 46(11):126–131, 2008.
- [6] S. Olariu and M.C. Weigle. *Vehicular Networks: from the theory to Practice*. Chapman and Hall/CRC, 2009.
- [7] D. Jiang, V. Taliwal, A. Meiei, W. Holfelder, and R. Herrtwich. Design of 5.9GHz DSRC-based Vehicular Safety Communication. *IEEE Wireless Communications*, 13(5):36–43, 2006.

- [8] F. Siewe, H. Zedan, and A. Cau. The Calculus of Context-aware Ambients. *Journal of Computer and System Sciences*, 77(4):597 – 620, 2011.
- [9] A. Ebner and H. Rohling. A self-organized radio network for automotive applications. In *proceedings of the conference of ITS 2001, 8th World Congress on Intelligent Transportation Systems*, 2001.
- [10] Fatal car accident. <http://www.car-accidents.com/pages/fatal-accident-statistics.html> Retrieved on 17.04.2010.
- [11] J. Li and C. Chigan. Achieving robust message dissemination in vanet: Challenges and solutions.
- [12] J. Zhao and G Cao. VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology*, 57(3):1910–1922, 2008.
- [13] V. Naumov and T. R. Gross. Connectivity-aware routing (car) in vehicular ad hoc networks. In *proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2007.
- [14] I. Leontiadis and C. Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In *IEEE International Symposium on a In World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2007.
- [15] Y. Lin, Y. Chen, and S. Lee. Routing protocols in Vehicular Ad Hoc Networks: A Survey and future perspectives. *Journal of Information Science and Engineering*, (4):1–20.
- [16] J. Lebrun, C. Chuah, D. Ghosal, and M. Zhang. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. In *In IEEE Vehicular Tech. Conference*, pages 2289–2293, 2005.
- [17] K. C. Lee and M. Lee, U. aqnd Gerla. Survey of Routing Protocols in Vehicular Ad Hoc Networks. *Pervasive and Mobile Computing*, 2009.

- [18] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto. GVGrid: A QoS Routing Protocol for Vehicular Ad Hoc Networks. In *proceedings of 14th IEEE International Workshop on Quality of Service (IWQoS)*, pages 130–139, 2006.
- [19] D. Bernsen, J. and Manivannan. Unicast Routing Protocols for Vehicular Ad Hoc Networks: A Critical Comparison and Classification. *Pervasive and Mobile Computing*, 5(1):1–18, 2009.
- [20] E. M. Royer and C. Toh. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communication*, 6(2), 1999.
- [21] C. S. Murthy and B. S. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, first printing edition, 2004.
- [22] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks journal*, 2(1), 2004.
- [23] R. E. Bellman. *Dynamic Programming*. Princeton University Press, 2004.
- [24] L. R. Ford and D. R. Fulkerson. *Flows in Network*. Princeton, N.J., Princeton University Press, 1962.
- [25] C. E. Perkins and T. J. Watson. Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers. In *proceedings of ACM SIGCOMM 1994 Conference on Communications Architectures*, UK, 1994.
- [26] S. Murthy and J.J. Garcia-Luna-Aceves. An Efficient Routing Protocol for Wireless Networks. *ACM/Baltzer Mobile Networks and Appli-*

- cations, special issue on Routing in Mobile Communications Networks*, 1(2), 1996.
- [27] S. Murthy and J.J. Garcia-Luna-Aceves. A Routing Protocol for Packet Radio Networks. In *Proceedings of the First Annual ACM International Conference on Mobile Computing and Networking*, pages 86–95, Berkeley, CA, USA, 1995.
- [28] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *proceedings of IEEE International Multi Topics Conference (INMIC)*, pages 62–68, 2001.
- [29] G. Pei, M. Gerla, and T. Chen. Fisheye state routing in mobile ad hoc networks. In *proceeding of Workshop on Wireless Networks and Mobile Computing (ICDCS)*, year = 2000, pages = 71–78.
- [30] G. Pei, M. Gerla, and T. Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 70–74, 2000.
- [31] J.J. Garcia-Luna-Aceves and M. Spohn. Source-tree routing in wireless networks. In *Proceedings of the 7th International Conference on Network Protocols (ICNP)*, pages 273–282, 1999.
- [32] T. Chen and M. Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. In *Proceedings of IEEE International Conference on Communications Conference (ICC)*, pages 171–175, 1998.
- [33] D. B. Johnson. Routing in ad hoc networks of mobile hosts. In *proceedings of the IEEE workshop on mobile computing systems and applications*, pages 158–163, 1994.

- [34] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.
- [35] D. Johnson, D. Maltz, and Y. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). In *IETF MANET Working Group, Internet Draft, draft-ietf-manet-dsr-10.txt*, 2004.
- [36] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications(WMCSA)*, pages 90–100, 1999.
- [37] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1405–1413, Kobe, Japan, 1997.
- [38] V. D. Park and M. S. Corson. Temporally-ordered routing algorithm (tora) version 1. In *Internet Draft, draft-ietf-manet-tora-spec- 03.txt, work in progress*, 2001.
- [39] C. Toh. Associativity-Based Routing for Ad-Hoc Networks. *Wireless Personal Communications Journal, Special Issue on Ad-Hoc Networks*, 17(8), 1999.
- [40] C. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *proceedings of IEEE 15th Annual International Phoenix Conference*, 1996.
- [41] R. Dube, C. D . Rais, K. Wang, and S. K. Tripathi. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. *IEEE Personal Communications*, 4:36–45, 1997.

- [42] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17:1439–1453, 1999.
- [43] Z. J. Haas, M. R. Pearlman, and P. Samar. The zone routing protocol (zrp) for ad hoc networks. In *draft-ietf-manet-zone-zrp-04.txt*, 2002.
- [44] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17:1454–1465, 1999.
- [45] R. Flury and R. Wattenhofer. Mls: an efficient location service for mobile ad hoc networks. In *In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. In MobiHoc 2006*, New York, NY, USA, 2006.
- [46] B. Karp. Gpsr: Greedy perimeter stateless routing for wireless networks. In *proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, 2000.
- [47] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto. A stable routing protocol to support its services in vanet networks. *IEEE Transactions on Vehicular technology*, 56(6):3337–3347, 2007.
- [48] C. Lochert, M. Mauve, Fubler, and H. Hartenstein. Geographic routing in City Scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 9(1):164–171, 2005.
- [49] H. Fubler, H. Hartenstein, J. Widmer, M. Mauve, and W. Effelsberg. Contention-based forwarding for street scenarios. In *Proceedings of the*

- 1st International Workshop in Intelligent Transportation (WIT 2004)*, pages 155–160, Hamburg, Germany, 2004.
- [50] A. Skordylis and N. Trigoni. Delay-bounded routing in vehicular ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 341–350, 2008.
- [51] S. Ahmed and S. S. Kanere. Skvr: Scalable knowledge-based routing architecture for public transport networks. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET’06*, pages 92–93, ACM, New York, NY, USA, 2006.
- [52] Y. Ding, C. Wang, and L. Xiao. A static -node assisted adaptive routing protocol in vehicular networks. In *Proceedings International Conference on Mobile Computing and Networking of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 59–68, Montreal, Quebec, Canada, 2007.
- [53] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, 2006.
- [54] S. Bai, Z. Huang, and J. Jung. Beacon-based cooperative forwarding scheme for safety applications in ivc. In *INC, IMS and IDC, 2009. NCM’09. Fifth International Joint Conference on*, pages 1449–1456. IEEE, 2009.
- [55] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 802.11*. 1999.

- [56] J. Jakubiak and Y. Koucheryavy. State of the art and research challenges for VANETs. In *proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pages 912–916, 2008.
- [57] A. K. Dey. Understanding and using context. *Journal of Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
- [58] J. Pascoe. Adding Generic Contextual Capabilities to Wearable Computers. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*, pages 92–99, 1998.
- [59] T. Rodden, K. Chervest, N. Davies, and A. Dix. Exploiting context in hci design for mobile systems. In *proceedings of Workshop on Human Computer Interaction with Mobile Devices*, 1998.
- [60] S. Poland. *Ubiquitous Computing: Smart Devices, Environments and Interactions*. Wiley, 2009.
- [61] S. Loke. *Context-Aware Pervasive Systems*. Auerbach Publications, 2006.
- [62] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad hoc and Ubiquitous Computing*, 2(4):263–277, 2007.
- [63] R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.
- [64] D. Sangiorgi and D. Walker. *PI-Calculus: A Theory of Mobile Processes*. Cambridge University Press, New York, NY, USA, 2001.
- [65] S. McCanne and S. Floyd. Network simulator. <http://www.isi.edu/nsnam/ns/>, KevinFall, KannanVaradhan, and the VINT project Retrieved on 15.05.2010.

- [66] Global mobile information system simulation glomosim. <http://pcl.cs.ucla.edu/projects/glomosim/> Retrieved on 21.05.2010.
- [67] Opnet modeler. <http://www.opnet.com/products/modeler/home.html> Retrieved on 04.06.2010.
- [68] K. Fall and K. Varadhan. Ns notes and documentation, the vint project, uc berkeley, lbl, usc/isi, and xerox parc. <http://www.isi.edu/nsnam/ns/ns-documentation.html> Retrieved on 16.05.2010.
- [69] M. M. Al-Doori, A. H. Al-bayatti, and H. Zedan. Context aware architecture for sending adaptive hello message in vanet. In *proceedings of Casemans 2010, the 4th ACM International Workshop on Context Awareness for Self-Managing Systems*, pages 26–29, 2010.
- [70] H. Hartenstein and K. Laberteaux. *VANET: Vehicular Applications and Inter-Networking Technologies*. John Wiley and Sons, 2009.
- [71] M. M. Al-Doori, F. Siewe, and A. H. Al-Bayatti. Vshdrp: Vehicle second heading directional routing protocol in vanet. In *proceeding of 4th IEEE International Conference on Computer Science and Information Technology (ICCSIT2011)*, Chengdu, China, 2011.
- [72] A. Festag, A. Hessler, R. Baldessari, L. Le, W. Zhang, and D. Westhoff. Vehicle-to-Vehicle and road-side sensor communications for enhanced road safety, 2009.
- [73] J. Luo and J. Hubaux. A survey of Inter-vehicle Communication. Technical report, School of Computer and Communication Sciences, EPFL, 2006.
- [74] Task Group p. Ieee 802.11p wireless access for vehicular environments. draft standard. <http://grouper.ieee.org/groups/802/11/> Retrieved on 30.01.2010.

- [75] S. Y. Wang, C. L. Chou, K. C. Liu, T. W. Ho, W. J. Hung, C. F. Huang, M. S. Hsu, H. Y. Chen, and C. C. Lin. Improving the channel utilization of IEEE 802.11P/1609 Networks. In *proceedings of the of Wireless Communications and Networking Conference WCNC*, pages 1–6. IEEE, 2009.
- [76] O. Shagdar, T. Ohyama, M. N. Shirazi, S. Tang, R. Suzuki, Miura R., and S. Obana. Message dissemination in inert-vehicle CDMA Networks for Safety driving support. In *Proceedings of the Vehicular Technology Conference(VTC)*, 2009.
- [77] A. T. Abd El-Nabi. *Modeling and simulation of routing protocol for ad hoc networks combining queuing network analysis and ANT colony algorithms*. PhD thesis, University of Duisburg-Essen, 2005.
- [78] M. Sarela. Measuring the Effects of Mobility on Reactive Ad Hoc Routing Protocols). Technical report, Helsinki University of Technology Laboratory for Theoretical Computer Science, Research Reports 91, 2004.
- [79] Defence advanced research project agency (dapra). <http://www.darpa.mil> Retrieved on 03.11.2009.
- [80] N. Abramson. Development of the ALOHANET. *IEEE Transactions on Information Theory*, 31(2):119–123, 1985.
- [81] Advanced research project agency (arpa). <http://searchnetworking.techtarget.com/definition/ARPA> Retrieved on 15.11.2009.
- [82] P. Mohapatra and S. V. Krishnamurth. *Ad Hoc Networks Technologies and Protocols*. Springer, first printing edition, 2004.
- [83] C. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice- Hall, 2002.

- [84] J. Al-Jaroodi. Security Issues In Wireless Mobile Ad Hoc Networks (MANET). Technical report, 2002.
- [85] Mobile ad hoc network. http://en.wikipedia.org/wiki/Mobile_ad_hoc_network Retrieved on 29.03.2010.
- [86] G. M. T. Abdalla, M. A. Abu-Rgheff, and S. M. Senouci. Current Trends in Vehicular Ad Hoc Networks. *Ubiquitous Computing and Communication Journal*, UbiRoads - Special Issue:263–277, 2008.
- [87] L. Wischhof. *Self-Organizing Communication in Vehicular Ad Hoc Networks*. PhD thesis, Technical University Hamburg-Harburg, 2007.
- [88] http://injuryprevention.bmj.com/content/8/suppl_4/iv26.full.pdf Retrieved on 20.01.2010.
- [89] H. Hartenstein and K.P. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–171, 2008.
- [90] W. Chen, R. K. Guha, T. J. Kwon, J. Lee, and I. Y. Hsu. A survey and challenges in routing and data dissemination in vehicular ad hoc networks. In *Proceedings of the 2008 IEEE International Conference on Vehicular Electronics and Safety*, pages 328–333, 2008.
- [91] M. Rudack and M. Meincke, M. and Lott. On the dynamics of ad hoc networks for inter vehicle communications (ivc), 2002.
- [92] S. Yousefi, M. S. Mousavi, and M. Fathy. Vehicular Ad Hoc Networks (VANETs): challenges and Perspectives. In *proceedings of 6th International Conference on ITS Telecommunications*, pages 761–766, 2006.
- [93] J. J. Blum, A. Eskandarian, and L. J. Hoffman. Challenges of Intervehicle Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):347–351, 2004.

- [94] M. M. Artimy, W. Robertson, and W. J. Phillips. Assignment of Dynamic Transmission Range Based on Estimation of Vehicle Density. In *proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 40–48, Cologne, Germany, 2005.
- [95] Traffic safety facts 2005 U.S. Department of Transportation. A compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimate system.
- [96] M. Nooij, B. Krosse, T. Broek, Maas. S., E. Nunen, and H. Zwijnenberg. Definition of necessary vehicle and infrastructure systems for Automated Driving. Technical report, European Commission DG Information Society and Media, 2010.
- [97] <http://en.wikipedia.org/wiki/3G> Retrieved on 25.01.2010.
- [98] A. Qureshi. Tavarua: Video Streaming with WWAN Striping. In *proceedings of the 14th annual ACM international conference on Multimedia*, pages 327–336, 2006.
- [99] S. J. Vaughan-Nichols. Mobile WiMax: The Next Wireless Battle Ground. *Computer*, 41(6):16–18, 2008.
- [100] S. J. Vaughan-Nichols. Achieving Wireless Broadband with WiMAX. *Computer*, 37(6):10–13, 2004.
- [101] WiMAX Forum. Mobile WiMAX - part I: A Technical Overview and Performance Evaluation. Technical report, 2006.
- [102] Its standards fact sheets. http://www.standards.its.dot.gov/fact_sheet.asp?f=66 Retrieved on 26.04.2010.
- [103] D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environment. In *proceedings of the Vehicular Technology Conference (VTC)*, pages 2036–240, 2008.

- [104] H. Moustafa and Y. Zhang. *Vehicular networks: techniques, standards, and applications*. CRC Press, 2009.
- [105] M. M. Artimy, W. Robertson, and W.J. Phillips. Connectivity in Inter-vehicle Ad hoc Networks . In *proceedings of Electrical and Computer Engineering*, pages 293–298, Niagara Falls, Canada, 2004.
- [106] M. M. Artimy, W. J. Phillips, and W. Robertson. Connectivity with Static Transmission Range in Vehicular Ad Hoc Networks. In *proceedings of 3rd Annual Communication Networks and Services Research Conference*, pages 237–242, Washington, DC, USA, 2005. IEEE Computer Society.

Appendix A

Overview in VANET

A.1 History and Background of MANET

Mobile ad hoc networks (MANET) have become increasingly popular and successful in the marketplace of wireless technology for the future, as indicated by the increasing use of Bluetooth and Wireless Local Area Networks (WLANs). The main opinion behind ad hoc networking is multi-hop relaying, the origin of which may be traced to 500 B.C. Darius I (533-486 B.C.), the king of Persia, invented an innovative communication system that was used to transmit messages and news from his capital to the remote provinces of his empire, by means of a line of shouting men positioned on tall structures or heights. This system was more than 23 times faster than normal messengers available at that time. The use of ad hoc voice communication was introduced in many ancient/ tribal societies with a string of repeaters of drums, trumpets or horns.

In 1970, DARPA (Defence Advanced Research Project Agency) [79] had a project called Packet Radio, whereby several wireless terminals could talk

with one another in a combat zone. Packet radio expands the concept of packet switching (developed from point-to-point communication networks) to the domain of broadcast radio networks. During the 1970s, a group of researchers headed by Norman Abramson (and others including N. Gaarder and N. Weldon) developed ALOHAnet [80], which connected the universities of the Hawaiian Islands together by means of a broadcasting property to send/receive packets of data in a single radio hop system. The ALOHA project led to the development of a multi-hop multiple-access packet radio network (PRNET) under the sponsorship of the Advanced Research Project Agency (ARPA), even though ALOHAnet was established for fixed single-hop wireless networks [81]. Unlike ALOHA, PRNET permits multi-hop communications over a wide geographical area, helping to establish the notion of ad hoc wireless networking in the same year [82].

A.2 MANET Characteristics

In recent years, the term *infrastructureless wireless networks* has become very common, leading researchers to investigate efforts to understand and develop this track of knowledge. MANET can be defined as a class of networks that does not require the wired access point and base stations as a support for intercommunication. A mobile ad hoc network is unlike a static network, as it has no infrastructure. It is a collection of mobile nodes in which communication is established in the absence of any fixed foundation. The multi-hop is the basis of the communication between remote nodes, since communication directly between neighbouring nodes is the only possible communication. These nodes are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. Each mobile

node acts as a host and a router, relaying information from one neighbour to another [20]. As shown in figure A.1, nodes *A* and *D* must enlist the aid of nodes *B* and *C* to relay packets between them in order to communicate.

MANETs have various defining characteristics that differentiate them from other wired and wireless networks, such as [83, 84, 21]:

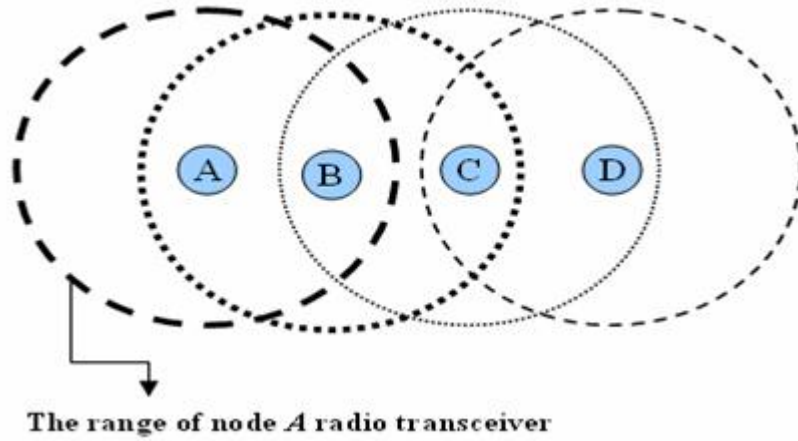


Figure A.1: Ad hoc network of four nodes, node *A* communicates with node *D*

- *Infrastructureless*: MANETs are formed based on the collaboration between independent, peer-to-peer nodes that wish to communicate with each other for a particular purpose. No prior organisation or base station is defined and all devices have the same role in the network. In addition, there are no pre-set roles, such as routers or gateways for the nodes participating in the network, unless specific arrangements are provided.
- *Dynamic Topology*: MANET nodes are free to move around; thus they could be in and out of the network, constantly changing its links and topology. In addition, the links between nodes could be bi-directional or unidirectional.
- *Low and Variable Bandwidth*: wireless links that connect the MANET

nodes have much smaller bandwidth than those with wires, while the effects of interference, noise and congestion are more visible, causing the available bandwidth to vary with the surrounding conditions and to be reduced even more.

- *Constrained Resources*: in general, most MANET devices are small hand-held devices, ranging from laptops, smartphones and personal digital assistants (PDA) down to cell phones. These devices have limited power (battery operated), processing capabilities and storage capacity.
- *Limited Device Security*: MANET devices are usually small and portable, and are therefore not restricted by location. As a result, these devices can easily be lost, damaged or stolen.
- *Limited Physical Security*: wireless links make MANET more susceptible to physical layer attacks, such as eavesdropping, spoofing, jamming and Denial of Service (DoS). However, the decentralised nature of MANETs makes them better protected against single failure points.
- *Short Range Connectivity*: MANET depends on radio frequency (RF) or infrared (IR) technology for connectivity, both of which are generally short range. Therefore, the nodes that wish to communicate directly need to be in close proximity to each other. To overcome this limitation, multi-hop routing techniques are used to connect distant nodes through intermediate nodes that act as routers.

Since ad hoc networks can be deployed rapidly without the support of a fixed infrastructure, they can be used in situations where temporary network connectivity is needed. Examples include conferences, meetings, crowd control, shared whiteboard applications (office workgroups), multi-user games, robotic pets, home wireless networks, office wireless networks, search and rescue, disas-

ter recovery and automated battlefields. These environments do not naturally have a central administration or infrastructure available.

A.3 Types of applications in MANET

Mobile ad hoc networks have been used in many applications to perform different tasks; these applications can be shown as follows [85]:

- *Wireless Sensor Networks (WSN)*: this type concentrates on monitoring physical actions or any change in environment status by exploiting the capability of autonomous sensors that are distributed in different locations to communicate wirelessly; it was first applied in a military application, and then it was used in other fields, such as industrial and civilian fields.
- *Internet-Based Mobile Ad Hoc Networks (IMANET)*: in this type, mobile nodes and fixed internet-gateway nodes are connected through ad hoc networks.
- *Vehicular Ad hoc Network (VANET)*: this is considered to be one of the MANET classes; this type of MANET emerged to provide a communication between neighbouring vehicles and between mobile vehicles and fixed units, such as roadside units (RSUs). The safety and comfortable applications for this type of MANET are considered as the core application. In the next sections, we will provide in detail the main definition of VANET and its types; the main characteristics and challenges will be identified, in addition to a detailed illustration of the class of its applications.

A.4 VANET Overview

Vehicle Ad hoc Networks (VANET) emerged as a subset of a Mobile Ad hoc Network (MANET) [18] application; it is considered as a substantial approach for the Intelligent Transportation System (ITS). In addition to non-safety applications [86] such as information and entertainment applications, VANETs were introduced to support drivers in the sense of safety issues and driving comfortably in an environment, as a step to construct a safer, cleaner and more intelligent environment [15] [72]. Recently, vehicles have been equipped with a sufficient number of sensors and devices, such as On Board Units (OBU); this combination enables the vehicles to sense the situations of other vehicles in the range of communication, through exploiting the infrastructures like the Road Side Unit (RSU) creating Vehicle to Infrastructure (V2I), or to interact directly with vehicles creating Vehicles to vehicle (V2V). Each vehicle can collect any information that is related to safety issues, such as warning messages for accident avoidance, traffic jams and collision situations, or other non-safety information such as a weather forecast, tourism information and Electronic Toll collection (ETC) [15]. At the same time, the vehicle can inform other vehicles about its current situation, such as its position and direction, which will provide valuable information that can be used to support their decision for the next action in terms of taking a particular lane and adjusting speed or direction [12].

A.5 Vehicle Communication Categories

Communication in vehicle ad hoc networks can be categorised into three main classes [87], as shown in figure A.2:

- (a) In-vehicle communications
- (b) Inter-vehicle communications
- (c) Vehicle-to-Infrastructure Communication

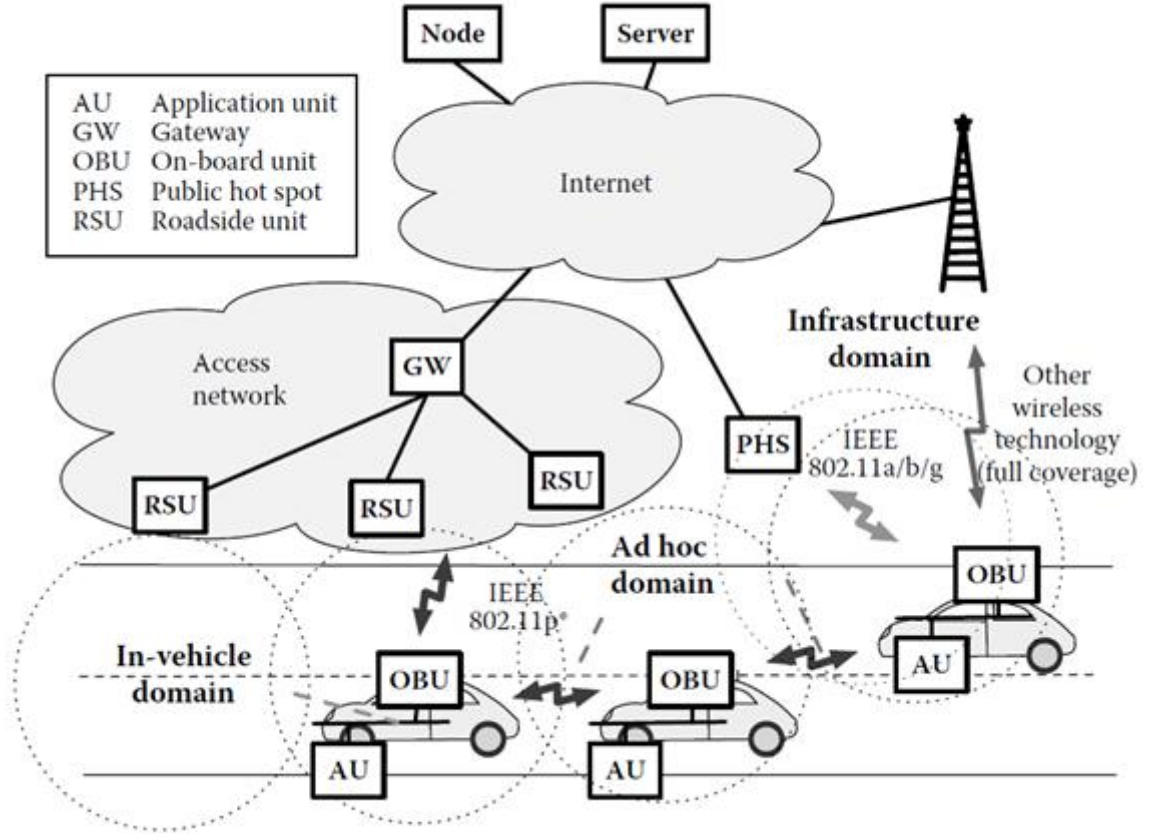


Figure A.2: Types of communication in VANET [3]

A.5.1 In-Vehicle Communications (InVC)

In-Vehicle Communication (InVC) contributes in exchanging information between OBU and Application Units (AU) in such a vehicle, which is common in modern cars. Generally, communications for in-vehicle areas have two applications: the first is the in-vehicle network of sensors, actuators and controllers,

and the second is high rate multimedia communication for comfort applications, (e.g. passenger entertainment).

Since in most situations the number of communicating entities tends not to change over the vehicle lifetime, the topology of in-vehicle communication networks is a stable topology; obviously, it defines a restricted group of possible communication partners and relies on wireline communication. Controller networks in particular have tight requirements on delay and integrity, whereas in the case of comfort applications, consequences of violations of the maximum allowed delay or data corruption are less serious, but higher data rates are required. The necessity of communication system standardisation for in-vehicle network comes from the high rate of increase in the number of integrated electronic components [88].

A.5.2 Inter-Vehicle Communication (IVC)

Many researchers in recent years have turned their attention to Inter-Vehicle Communication (IVC), particularly in the USA, EU and Japan, owing to its ability to expand the driver's prospects which leads, as a result, to enhancing the safety issues relating to road traffic and increasing the efficiency of system safety [73]. In this type of communication, vehicles are managed in a decentralised manner, through allowing the vehicle to initiate direct communication with other vehicles without using any support from infrastructures, V2V communication can act as a crucial building block of an intelligent transport system (ITS). It is considered to be a more realistic solution if infrastructures are unavailable [76] and if the high cost of installing and deploying any type of these infrastructures is taken into account.

Microwaves are considered to be the most important stream used in IVC; in particular, the Dedicated Short Range Communication (DSRC) which was allocated by the FCC (Federal Communication Commission) in the U.S. utilises spectrum over 75 MHz in the 5.9 GHz band, while in the EU and Japan the 5.8 GHz band is used [89].

As shown in figure A.3, vehicle *A* broadcasts information such as direction, location and speed to other vehicles within its range of transmission at time t , while vehicle *B* in the opposite direction will receive *A*'s information, which can rebroadcast this information with *B*'s information. After a period of time $t + d$, another vehicle like vehicle *C* can receive this information; thus vehicle *C* will have the predicted information regarding its traffic conditions [4].

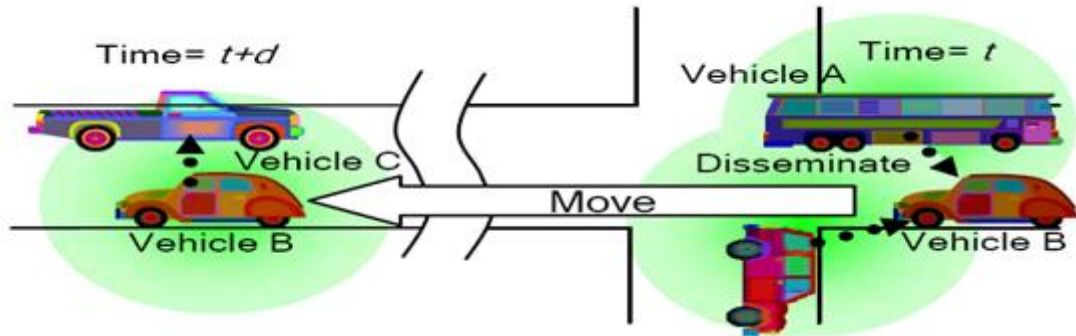


Figure A.3: Inter-vehicle Ad hoc (V2V) communication type [4]

A.5.3 Vehicle-to-Infrastructure Communication

Vehicle-to-Infrastructure Communication (V2I) also called Road-to-Vehicle Communication (RVC), is considered to be an expensive alternative, owing to its requirement of a high number of terminals, such as road side units and

Base Stations (BS), which are considered as the coordinator of each connection between subscribers in cellular networks, to ensure a complementary cover to all roads [72]. The base station here plays the main role in controlling the negotiation process and setting up the connection between the infrastructures and mobile vehicle.

As illustrated in figure A.4, V2I provides a facility that enable vehicles to initiate a connection with fixed units that are distributed along the road, either by disseminating useful information that describes the vehicle's current situation, which can be used to support the decision that can be taken by other vehicles in terms of movement action, direction and speed, or by receiving this information to avoid taking a certain action, such as ETC and weather status. These units are known as Roadside Units (RSUs) which employs the Wireless Access in Vehicular Environment (WAVE) standard [90], which uses DSRC to communicate with vehicles; furthermore, V2I exploits a cellular network, e.g. GSM, UMTS, or WiMAX, to make bidirectional communications and provide internet access and infotainment applications.

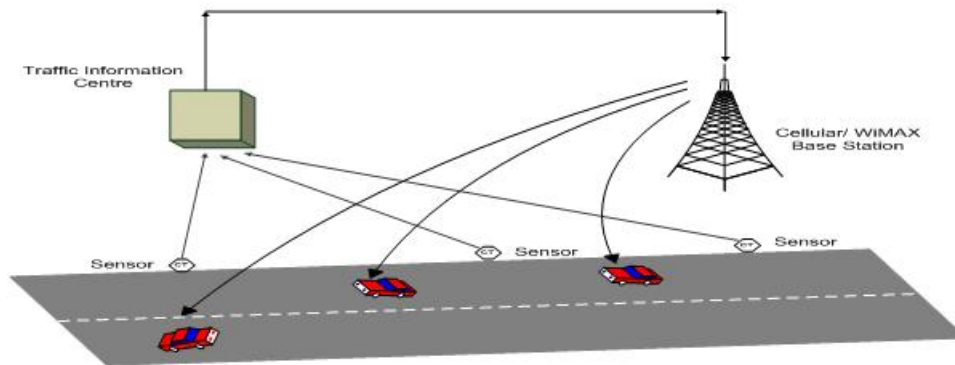


Figure A.4: V2I communication type

A.6 VANET Characteristics

It is noticeable that VANET differs from MANET in many ways when a comparison is made with the traditional aspects of MANET; these differences are represented by the difficulties that faced VANET in attempting to apply the conventional protocols and communication paradigms. Also, as mentioned before in section 2.6, VANET is considered to be a special case of MANET; therefore it is vital to go through the general aspects of VANET [91].

VANET can be characterised according to the following features:

- *Mobility*: because of the high travelling speed of vehicles, especially on highway roads, this may reach 70 m/h . A continuous and rapid change in network topology can occur. At the same time, since vehicles follow a certain mobility pattern that is a function of the underlying roads, the traffic lights, the speed limit, traffic condition, and drivers' driving behaviours [56], which means that vehicle movement is restricted and can be predicted in certain ways. Moreover, vehicle drivers can react according to the content of the messages that are received throughout the network, which will cause a change in its behaviour which will also be reflected on the network topology [92].

The assumption that vehicles move in the same direction when they travel on the same road can support the concept of prediction nature of VANET [93], in addition to taking the digital map utilities into consideration [48]. All these factors assist in eliminating the change in network topology and make it more stable.

VANETs depend on ad hoc inert-vehicle communication type when it

elects to communicate directly for exchanging information without the use of any infrastructures. A high time and space variant in mobile radio channels is the result of a high mobility situation; in this case, an adaptive and appropriate physical and medium access layer is the requirement for realising a reliable communication [93] [94] [91].

- *Power Management*: saving energy is considered to be one of VANET features; it is not an issue in VANET, unlike MANET which considers power consumption as a critical challenge. Power in vehicles is reserved much more highly than in a classical mobile node; in other words, vehicles have the ability to provide sufficient power for a long life time, as a result of the on-board batteries that can be used to support the need for power, and which can be recharged when necessary [56].
- *Large Computational Ability*: the advantage of the large size and weight for vehicles compared with conventional mobile nodes is that it support vehicles with powerful computers, large equipment and extremely high memory capacity, which helps in supporting the computational ability of equipped devices (e.g. component of in-board unit and sensors) [92] and saving valuable information, which assists in enhancing the efficiency of the system in terms of data dissemination and the selection of the appropriate intermediate vehicles which help in supporting the routing protocol. Furthermore, having sufficient size and weight can enhance the security issues in VANET by supporting a high degree of cryptography, which leads to obtaining more trusted information in the network [6].
- *Connectivity*: increasing the guarantee of connectivity is considered to be an essential factor that improves the system's reliability; on one hand, this can be accomplished by exploiting the Vehicle-to-Infrastructures communication type (V2I) represented by roadside units which use DSRC and

cellular networks such as GSM and WiMAX technology which improve the work of Vehicle-to-Vehicle types (V2V). On the other hand, V2V communication types have reached high stages of development in order to achieve the task of connectivity improvement. Authors in [90] mentioned that disseminating data in a multi-hop technique provides an extension to the range of transmission, and supports the guarantee to deliver emergency and warning messages especially for safety applications. Generally, the requirement of multi-hop data dissemination can be accomplished by having the node location information and forwarding packet method; two types of technology can achieve that, either by a routing protocol or by using a method of forwarding with knowledge of the location service.

- *Safety Applications*: in addition to the comfort (non-safety) applications, the road safety issues and transportation efficiency represent the main goal behind the evolution of VANET; for that reason safety applications can be characterised as one of the essential features of VANET, the noticeable increase in fatalities statistics, since vehicles crashes have an impact on finding a valuable solution to improve the safety property. One of the major solutions is, for instance, that it can help in avoiding accident occurrences, by delivering the warning messages to the desirable region as quickly as possible; that can be achieved through enhancing the process of data dissemination in terms of routing protocols, as indicated in figure A.5, vehicles approaching the curve are warned by the vehicle in the curve (using V2V) and by exploiting the RSU located in the curve (using I2V).
- *Cost Effective*: in addition to reducing the number of injured people caused by accidents, which has an effect on healthcare cost, it is vital to argue the advantages that can be gained through focusing the work of

VANET in Vehicle-to-Vehicle communication types; this type of communication is considered to be very suitable for end-users in terms of cost effectiveness, owing to its aspect of depending on V2V communication without use of any infrastructure. These infrastructures such as (RSU) cellular networks are already considered to be an expensive alternative for end users.

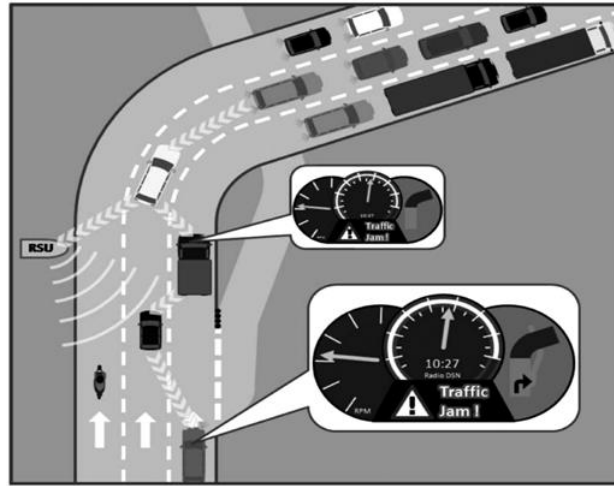


Figure A.5: Avoiding crashes with the aid of V2V and V2I [5]

A.7 Applications in VANET

VANET has a large and variety number of applications, exploiting the combination of information that is collected from the inside and outside the vehicle through utilising devices which are fitted in vehicles, such as radar and wireless sensors [73], in addition to integrating the fixed infrastructures as cellular networks and WiMAX technology. Moreover the focus of this thesis is on increasing the efficiency of safety applications, which can be improved by developing a new routing protocol that support this type of application.

Applications in VANET can be classified into two main categories:

- Entertainment/ Information Applications
- Safety Applications

A.7.1 Entertainment/ Information Applications

The information/entertainment applications, sometimes called non-safety applications, focus on leveraging the efficiency of traffic and providing drivers with more comfortable journeys, for instance, informing the passenger where the nearest fuel station is, supplying information about the current weather condition in a specific area, illustrating restaurant menus and their prices [92], giving information about the availability of the parking spaces in the nearest car parks, providing the facilities to access internet services in a smooth way during the journey and enabling passengers to play games online [70].

A.7.2 Safety Applications

The U.S. Department of Transportation (U.S. DOT) mentioned that [95], in 2005, the number of people injured and killed in car crashes had increased, which led to more money being spent on healthcare, rising to billions of dollars; at the same time, the rapid growth and improvement in the field of wireless communications turned the attention of governments to vehicle manufacturers, especially in the USA, the EU and Japan, to investigate this technology in order to support road safety and present a clean environment in terms of avoiding vehicle accidents.

DSRC is the major protocol on which safety applications are based; it works in the 5.9 GHz band in the USA, while in the EU and Japan, the 5.8 GHz

is the dedicated frequency band. Two types of communication are provided by DSRC: Vehicle-to-Vehicle (V2V) provides the facility to exchange data between vehicles, and Vehicle-to-Infrastructure (V2I) which enables the vehicle to communicate with roadside units (RSUs). The DSRC spectrum comprises seven channels; each channel width is 10MHz, and channel 178 is particularly constrained for safety communications only, as illustrated in section 2.9. According to the functions applied by the safety applications, information can be acquired through sensors fitted in the vehicle from other vehicles in the network or together.

In each application, data are disseminated after they have been processed to their destination, which can be another vehicle or infrastructure, depending on the reason and conditions to send such a message; two types of messages can be used in safety applications: periodic messages and event-driven messages.

- Periodic Messages: vehicle-related information such as direction, position and speed, need to be known from other surrounding vehicles to take it into consideration in order to make a decision in an attempt to prevent a hazardous situation from happening. Therefore, periodic messages are considered to be an important type of message that supports the decision that can be taken in safety applications; however, it may lead to wasted bandwidth consumption, especially in dense environments, in addition to increasing the probability of a storm problem occurring.
- Event-Driven Messages: this type of message is disseminated only when hazardous conditions are occurring, otherwise this type of messages is not sent. It occupies a high level of priority; the main difficulty in this type of message is the necessity to increase the guarantee of delivering

this message to all vehicles that need to know about it.

The limited capabilities of current safety technology stand as an obstacle; for example, one needs a direct line of sight (LOS) before introducing valuable and trusted safety applications. All of that is pushing the researchers, with the support of governments and vehicle manufacturers, to improve this technology in terms of achieving high levels of safety. A combination between infrastructure and vehicle communications has been done by the Vehicle Infrastructure Integration (VII) project [96]; this project is a union between U.S. DOT and ten departments of transportation (DOTs) and manufacturers of automotive. This union aims mainly to leverage Safety levels by providing the roads and new vehicles with communication technology.

A.8 Wireless Access Technology in VANET

The availability of numerous wireless access technologies has an impact on providing an appropriate air interface in order to accomplish the process of data dissemination. In general, the concept of employing these wireless access standards has been achieved by the use of two approaches: the first relies on a centralised management that utilises a coordinator which controls the connections between all nodes (vehicles) in the network; this approach has a high guarantee for delivering data to its destination. In contrast, the second approach is concerned with initiating a direct communication (distributed communication) between any two nodes; in other words, no dominant coordinator can be used) [73]. These technologies may be illustrated as follows:

- *Cellular Networks (2/3G)*: owing to the growing need for mobile networking, a fast development in cellular network has been occurring. GSM,

which emerged as 2G technology, focused on achieving reliable and secure connections [56] that provide a maximum data rate of 9.6 kb/sec. Systems depending on GSM technology utilise EDGE with up to 384 kb/sec and GPRS with up to 171kb/sec. IS-95 is another 2G technology which can provide up to 141 kb/sec. Consequently, the increasing need for a high data rate as a step to support the multimedia issues led to the appearance of 3G technology [97] such as UMTS/HSDPA, which can provide up to 2Mb/sec, and CDMA2000, with up to 3Mb/sec for downlink and 1.8 Mb/sec for uplink.

The performance of 3G technology in vehicular ad hoc networks can be estimated according to round trip delay (RTT) and disconnected lifetime Download/ upload throughput variations and the correlation between throughput and vehicle's speed [98].

- *WiMAX(IEEE802.16e / IEEE802.16m)* technology: the demand for a high data rate and reliable Wireless Broadband Access (WBA) system has been growing in the last few years, in addition to the essential demand for using this service in applications that need a high Quality of Service (QoS), especially for applications related to multimedia [99] and voice over IP (VoIP). WiMAX (Worldwide Interoperability for Microwave Access) has been developed as an evaluable solution that aims to satisfy these requirements (high data rate and reliable connection). The WiMAX system depends mainly on the standard of IEEE 802.16 in its general structure [100].

WiMAX comes with improving a set of issues in terms of WBA systems, with a set of crucial features which led to categorising the WiMAX in the

top list of Wireless Broadband Access systems; the first of those features is the high data rate which it promises through using the MIMO (Multi Input Multi Output) with OFDMA (Orthogonal Frequency Division Multiple Access) technology which gave the system more flexibility in uplink (UL) rate and download rate (DL); second, the quality of service that the standard of IEEE 802.16 takes into account from the beginning by classifying the connections into five classes: Unsolicited Grant Service (UGS), real time Polling Service (rtPS), non real time Polling Service (nrtPS), extended real time Polling Service (ertPS) and Best Effort (BE); third, the mobility which can be accomplished through supporting the real time applications by WiMAX system by introducing a handoff mechanism with less than 50 millisecond latencies such as voice over IP (VoIP) applications; finally, WiMAX provides a wide range of coverage, which can reach a radius of up to 7 Km with a high capacity [101].

Consequently, the aim of developing the IEEE802.16m standard is to build a new standard that meets the requirements of International Telecommunication Union-radio communication/International Mobile Telecommunication (ITU-R/IMT-advanced), which will lead to categorising the WiMAX technology under the 4G- next generation mobile networks operators; in addition to that, it aims to meet the cellular network requirements in backward compliant with the legacy system.

- *Wireless Local Area Network (WLAN)*: wireless local area network, or even Wi-Fi in its standard 802.11a/g, which grants 54 Mb/s with a minimum transmission range of 38 m for indoor use, reaching 140 m for outdoor use.
- *Combined Wireless Access Method*: many scenarios can arise in an at-

tempt to create a vehicular ad hoc network; in addition to vehicle-to-infrastructure (V2I), vehicle-to-vehicle (V2V), the DSRC technology can be exploited as a combination of V2I and V2V, as shown in figure A.6. CALM M5 accomplished the task of mixing a set of technology into a uniform and single formation, by adapting the 802.11p through adding a number of interface protocols which support the standard of GSM/HSCSD/GPRS (2/2.5G) and UMTS for 3G, infrared communication and wireless system in 60 GHz band [56].

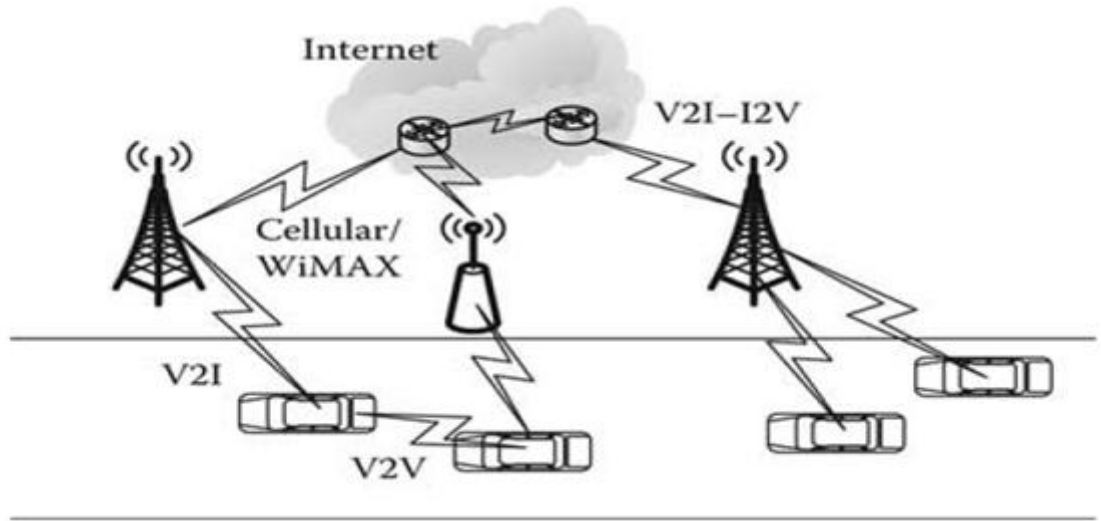


Figure A.6: Scenario of combined types of communications [6]

- *DSRC/WAVE*: the IEEE 802.11p standard [74] was created based on 802.11a; the dedicated short range communication (DSRC) is considered to be a communication technology for short and medium range [102]. The communications between high-speed vehicles travelling at speeds of up to 70 m/h or between a vehicle and roadside infrastructure unit with an average range of transmission of 300 m (up to 1000 m) have been established by the WAVE (Wireless Access in Vehicular Environments). A 75 MHz in 5.9 GHz band from the spectrum was allocated by the Federal Communication Commission (FCC) for the U.S. in 1999. However,

in Europe, 2.4 GHz was chosen first, before being changed to 5.8 GHz band, which is the same band as that used in Japan [73].

In addition, safety application is the essential aim of initiating the DSRC technology in terms of reducing fatalities and improving traffic flow. The IEEE 802.11p is the base of DSRC radio technology; it has been adjusted to reduce the overhead in the spectrum of DSRC. This standard is called the IEEE 1609 family, which was designed for management issues, security issues, networking services and multi-channel operations [75].

The DSRC spectrum comprises seven channels, as shown in figure A.7; 10 MHz is the width of each channel, channel 178 is the Control Channel (CCH) that was allocated for the purpose of safety communication, the two channels in the end of the spectrum were allocated for special use, while other channels (SCH) in the spectrum band are for safety and non-safety applications. Using the DSRC band is free according to the FCC; however, it is a licensed band and should not be confused with free used and unlicensed bands in 900 MHz , 2.4 GHz and 5 GHz , which is used with Wi-Fi, Bluetooth [103] [7] [75].

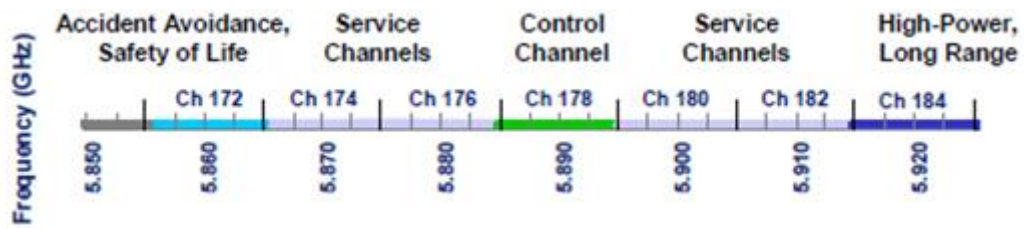


Figure A.7: DSRC channels spectrum [7]

A.9 Challenges and Requirements in VANET

Many issues arise when efforts are gathered towards running vehicular ad hoc networks in an attempt to provide an improvement to driver behaviour, with the aim of reducing the number of fatalities caused by automobile accidents. To realise the requirements that needed to deploy VANET concept, many factors that have a critical impact on achieving the VANET goal need to be taken into consideration, represented by safety applications and non-safety applications. Thus it is vital to specify the main important challenges in VANET [104], and the key challenges from the technical perspectives are as follows:

- *Signal fading*: objects placed as obstacles between two communicating vehicles are one of the challenges that can affect the efficiency of VANET; these obstacles can be other vehicles or buildings distributed along roads especially in the cities. Their impact is placed on preventing the signal from reaching its destination and increasing the fading in the transmitted signal [89].
- *Mobility and rapid change of topology*: continuous change in network topology is considered to be one of the most crucial problems in vehicular ad hoc networks, which depend on the high mobility of the vehicles owing to the high velocity of these vehicles, scalability requirements and variations in environment conditions [56].
- *Bandwidth limitations*: another key issue in the VANET is the absence of a central coordinator that controls the communications between nodes, and which has the responsibility of managing the bandwidth and contention operation. Therefore it is necessary to utilise the availability of bandwidth efficiently. There is a high probability that channel conges-

tion can occur, owing to the limited range of bandwidth frequency (10-20 MHz) for VANET applications, particularly in a high density environment. The fair use of bandwidth has its impact on reducing the time delay for disseminating messages; if a vehicle needs to send a message and finds there are no opportunities for transmission, it must wait for a time to have a chance for transmission, which will have an effect on increasing the latency, especially in urban areas and with the increase in the types of application in VANET [70].

- *Connectivity*: owing to the high mobility and rapid changes of topology, which lead to a frequent fragmentation in networks, the time duration required to elongate the life of the link communication should be as long as possible. This task can be accomplished by increasing the transmission power; however, that may lead to throughput degradation. Accordingly, connectivity is considered to be an important issue in VANET, although many studies in MANET [93, 105, 106] have focused on solving this problem. Nevertheless, it still occupies a wide portion of the efforts gathered towards developing VANET.
- *Small effective Diameter*: Owing to the small effective network diameter of a VANET, that lead to a weak connectivity in the communication between nodes. Therefore, maintaining the complete global topology of the network is impracticable for a node. The restricted effective diameter results in problems when applying existing routing algorithms to a VANET [70].
- *Security and privacy*: keeping a reasonable balance between the security and privacy is one of the main challenges in VANET; the receipt of trustworthy information from its source is important for the receiver. However, this trusted information can violate the privacy needs of the

sender [89].

- *Routing protocol*: because of the high mobility of nodes and rapid changes of topology, designing an efficient routing protocol that can deliver a packet in a minimum period of time with few dropped packets is considered to be a critical challenge in VANET. Furthermore, many researchers have concentrated on designing a routing protocol suitable for dense environments that have a high density of vehicles with close distances between them. Designing an efficient routing protocol has an impact on improving many factors; the first of these is enhancing the reliability of the system by leveraging the percentage of packets delivery, and secondly by reducing the extent of interference caused by high buildings in the city environment; the third factor is that taking scalability into consideration is essential to avoid conflict, if a simultaneous operation of unicast routing request has been initiated. Another factor is to deliver a packet in the shortest possible time, especially in the emergency situation; this factor is considered to be a very critical factor [15, 90, 89, 92, 5, 17, 104].

Appendix B

Formal specification in the Calculus of Context-aware Ambients(CCA) for VSHDRP1

This Appendix contains the full formal executable specification of the proposed routing protocol VSHDRP1. The specification contained three scenarios, namely *scenario1*, *scenario2* and *scenario3*.

Primary ambients:

- SN: source node
- DN: destination node
- IN1: intermediate node 1
- IN2: intermediate node 2

Directions:

- Direction 1: direction towards the destination node.

- Direction 2: opposite direction to direction 1.

Each node N contains the (secondary) ambients:

- NH1: contains the ids of all the neighbouring nodes, closer to the destination node, that are moving in direction 1.
- NH2: contains the ids of all the neighbouring nodes, closer to the destination node, that are moving in direction 2.
- Hspeed: contains the ids of all the neighbouring nodes moving in high speed.
- Lspeed: contains the ids of all the neighbouring nodes moving in low speed.
- SHD: contains the ids of all the neighbouring with SHD.

Predicates:

- hasNb(n): n is a neighbour closer to destination (regardless its direction).
- hasNb1(n): n is a neighbour closer to destination in direction 1.
- hasNb2(n): n is a neighbour closer to destination in direction 2.
- noNb1(): no neighbours in direction 1.
- hasSHD(n): n is a neighbour with SHD.
- noSHD(): no neighbours have SHD.
- highSpeed(n): n is a neighbour with high speed.
- lowSpeed(n): n is a neighbour with low speed.
- highestSpeed(n): n is a neighbour with the highest speed.
- hasNb1ofNb2(n): n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1.

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
SCENARIO I
-----

DN is neighbour to SN.

*/BEGIN_DECLS
  def has(n) = this | n[true] | true
  def at(n) = n[next (this | true)] | true
  def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
  def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
  def hasNb(n) = hasNb1(n) or hasNb2(n)
  def highSpeed(n) = somewhere(this | Hspeed[true| n[true]] | true)
  def lowSpeed(n) = somewhere(this | Lspeed[true| n[true]] | true)
  def highestSpeed(n) = highSpeed(n) or (not (somewhere(this | Hspeed[true | next true] | true))
and lowSpeed(n))
  def hasSHD(n) = somewhere (this | SHD[ true | n[true]] | true)
  def hasNbofNb1(n) = hasNb2(n) and somewhere(n[ true | NH1[true | next true]] | true)
  def noNb1() = not somewhere(this | NH1[true| next true] | true)
  def noSHD() = not somewhere(this | SHD[true| next true] | true)
//display code
display congruence
END_DECLS
SN[ // source node

/* BEGIN NODE BEHAVIOUR */
/*
  if DN is neighbour to SN then send message to DN.
*/
  <hasNb(DN)> DN::send(SN, hello).DN::recv().0
  |
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1 with SHD and highest speed.
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD(n) and highestSpeed(n) for n::send(SN, DN, hello).
n::recv().0 // DN is not neighbour to SN
  |
/*
  if no such intermediate nodes, look for an intermediate node in direction 2 which has a neighbour
in direction 1 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD()) and hasNbofNb1(n) for n::send(SN, DN, hello).n::recv().0
```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
// DN is not neighbour to SN

/* END NODE BEHAVIOUR */
|

  NH1[ // neighbourhood of the source node in direction 1 closer
// to the destination
    DN[0]
    | IN1[0]
  ]
|
  NH2[ // neighbourhood of the source node in direction 2 closer
// to the destination
    0
  ]
|
  Hspeed[ // neighbours in high speed
    0
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[ // neighbour with SHD
0
  ]
]

|

DN[ // destination node
  ::recv(s,msg).s::send().0
|
  NH1[ // neighborhood of the destination node in direction 1
    0
  ]
|
  NH2[ // neighborhood of the destination node in direction 2
    0
  ]
]
```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
|
  Hspeed[ // neighbours in high speed
    0
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[
0
  ]
]

|

IN1[ // intermediate node 1
  ::rcv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
  if destination node d is neighbour to IN then send message to d.
*/
  <hasNb1(d)>d::send(IN1, msg).d::rcv().0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1 with SHD and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
  for n::send(IN1, d, msg).n::rcv().0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n) for n::send(IN1, d, msg).n::rcv().0

|
  NH1[ // neighborhood of the intermediate node 1 in the direction 1
    IN2[0]
  ]
]
```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
|
  NH2[ // neighborhood of the intermediate node 1 in the direction 2
    0
  ]
|
  Hspeed[ // neighbours in high speed
    IN2[0]
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[
IN2[0]
  ]
]

|

IN2[ // intermediate node 2
  ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
  if destination node d is neighbour to IN then send message to d.
*/

      <hasNb1(d)>d::send(IN2, msg).d::recv().0

/*
  if d is not neighbour to IN then look for an intermediate node in direction 1 with SHD and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
  for n::send(IN2, d, msg).n::recv().0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 closer to d.
*/

  |find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n) for n::send(IN2, d, msg).n::recv().0
```

```

|
  NH1[ // neighborhood of the intermediate node 2 in the direction 1
    IN3[0]
  ]
|
  NH2[ // neighborhood of the intermediate node 2 in the direction 2
    0
  ]
|
  Hspeed[ // neighbours in high speed
    IN3[0]
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[
IN3[0]
  ]
]

|

IN3[ // intermediate node 3
  ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
  if destination node d is neighbour to IN then send message to d.
*/

  <hasNb1(d)>d::send(IN3, msg).d::recv().0

/*
  if d is not neighbour to IN then look for an intermediate node in direction 1 with SHD and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(IN3, d, msg).n::recv().0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 closer to d.
*/

```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
| find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
for n::send(IN3, d, msg).n::recv().0

|
  NH1[ // neighborhood of the intermediate node 3 in the direction 1
    DN[0]
  ]
|
  NH2[ // neighborhood of the intermediate node 3 in the direction 2
    0
  ]
|
  Hspeed[ // neighbours in high speed
    0
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[ // can be "one" or "zero".
0
  ]
]
```

SCENARIO II

In this scenario the destination node DN is not neighbour to the source node SN.
Rather the neighborhood relation is defined as follows:

SN--->IN1--->IN2--->IN3--->DN

BEGIN_DECLS

```
def has(n) = this | n[true] | true
def at(n) = n[next (this | true)] | true
def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
def hasNb(n) = hasNb1(n) or hasNb2(n)

def highSpeed(n) = somewhere(this | Hspeed[true| n[true]] | true)

def lowSpeed(n) = somewhere(this | Lspeed[true| n[true]] | true)

def highestSpeed(n) = highSpeed(n) or (not (somewhere(this | Hspeed[true | next true] | true))
and lowSpeed(n))

def hasSHD(n) = somewhere (this | SHD[ true | n[true]] | true)

def hasNbofNb1(n) = hasNb2(n) and somewhere(n[ true | NH1[true | next true]] | true)

def noNb1() = not somewhere(this | NH1[true| next true] | true)

def noSHD() = not somewhere(this | SHD[true| next true] | true)

//display code
display congruence
END_DECLS
SN[ // source node

/*
  if DN is neighbour to SN then send message to DN.
*/
<hasNb(DN)> DN::send(SN, hello).DN::recv().0
|
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD and highest speed.
*/
find n: not hasNb(DN) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(SN, DN, hello).n::recv().0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 closer to DN.
*/
find n: not hasNb(DN) and (noNb1() or noSHD()) and hasNbofNb1(n)
for n::send(SN, DN, hello).n::recv().0 // DN is not neighbour to SN
|

NH1[ // neighborhood of the source node in direction 1 closer
// to the destination
  IN1[0]
]
|
NH2[ // neighborhood of the source node in direction 2 closer
// to the destination
```



```

        0
    ]
|
    Hspeed[ // neighbours in high speed
        IN1[0]
    ]
|
    Lspeed[ // neighbours in low speed
        0
    ]

| SHD[ // neighbour with SHD
IN1[0]
    ]
]

|

DN[ // destination node
    ::recv(s,msg).s::send().0
|
    NH1[ // neighborhood of the destination node in direction 1
        0
    ]
|
    NH2[ // neighborhood of the destination node in direction 2
        0
    ]
|
    Hspeed[ // neighbours in high speed
        0
    ]
|
    Lspeed[ // neighbours in low speed
        0
    ]

| SHD[
0
    ]
]

```

```

|

IN1[ // intermediate node 1
    ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
    if destination node d is neighbour to IN then send message to d.
*/
    <hasNb1(d)>d::send(IN1, msg).d::recv().0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD and highest speed.
*/

    | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
    for n::send(IN1, d, msg).n::recv().0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 closer to d.
*/
    | find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1ofNb1(n)
    for n::send(IN1, d, msg).n::recv().0

|

    NH1[ // neighborhood of the intermediate node 1 in the direction 1
        IN2[0]
    ]
|
    NH2[ // neighborhood of the intermediate node 1 in the direction 2
        0
    ]
|
    Hspeed[ // neighbours in high speed
        IN2[0]
    ]
|
    Lspeed[ // neighbours in low speed
        0

```

```

    ]

    | SHD[
IN2[0]
    ]
]

|

IN2[ // intermediate node 2
    ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
    if destination node d is neighbour to IN then send message to d.
*/
* /
    <hasNb1(d)>d::send(IN2, msg).d::recv().0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD and highest speed.
*/

    | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(IN2, d, msg).n::recv().0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 closer to d.
*/
* /
    | find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
for n::send(IN2, d, msg).n::recv().0

|

    NH1[ // neighborhood of the intermediate node 2 in the direction 1
        IN3[0]
    ]
|

    NH2[ // neighborhood of the intermediate node 2 in the direction 2
        0
    ]
|

```

```

    Hspeed[ // neighbours in high speed
        IN3[0]
    ]
|
    Lspeed[ // neighbours in low speed
        0
    ]

| SHD[
IN3[0]
    ]
]

|

IN3[ // intermediate node 3
    ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
    if destination node d is neighbour to IN then send message to d.
*/
    <hasNb1(d)>d::send(IN3, msg).d::recv().0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD and highest speed.
*/

    | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(IN3, d, msg).n::recv().0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 closer to d.
*/
    | find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
for n::send(IN3, d, msg).n::recv().0

|
    NH1[ // neighborhood of the intermediate node 3 in the direction 1
        DN[0]
    ]

```

```

    ]
  |
    NH2[ // neighborhood of the intermediate node 3 in the direction 2
      0
    ]
  |
    Hspeed[ // neighbours in high speed
      0
    ]
  |
    Lspeed[ // neighbours in low speed
      0
    ]
  ]

  | SHD[ // can be "one" or "zero".
0
  ]
]

/*
SCENARIO III

In this scenario the destination node DN is not neighbour to the source node SN.
Rather the neighborhood relation is defined as follows:

SN--->IN1--->IN2--->IN3--->DN

But IN1 and IN3 are in direction 2.

*/

BEGIN_DECLS
  def has(n) = this | n[true] | true
  def at(n) = n[next (this | true)] | true
  def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
  def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
  def hasNb(n) = hasNb1(n) or hasNb2(n)
  def highSpeed(n) = somewhere(this | Hspeed[true| n[true]] | true)
  def lowSpeed(n) = somewhere(this | Lspeed[true| n[true]] | true)
  def highestSpeed(n) = highSpeed(n) or (not (somewhere(this | Hspeed[true | next true] | true))
and lowSpeed(n))

```

APPENDIX B. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP1

```
def hasSHD(n) = somewhere (this | SHD[ true | n[true]] | true)
def hasNbOfNb1(n) = hasNb2(n) and somewhere(n[ true | NH1[true | next true]] | true)
def noNb1() = not somewhere(this | NH1[true| next true] | true)
def noSHD() = not somewhere(this | SHD[true| next true] | true)

//display code
display congruence
END_DECLS

SN[ // source node

/*
  if DN is neighbour to SN then send message to DN.
*/
  <hasNb(DN)> DN::send(SN, hello).DN::recv().0
  |
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
  with SHD and highest speed.
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
  for n::send(SN, DN, hello).n::recv().0 // DN is not neighbour to SN
  |
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD()) and hasNbOfNb1(n)
  for n::send(SN, DN, hello).n::recv().0 // DN is not neighbour to SN
  |

  NH1[ // neighborhood of the source node in direction 1 closer
// to the destination
    0
  ]
  |
  NH2[ // neighborhood of the source node in direction 2 closer
// to the destination
    IN1[0]
  ]
  |
  Hspeed[ // neighbours in high speed
    IN1[0]
```

```

    ]
  |
    Lspeed[ // neighbours in low speed
      0
    ]

  | SHD[ // neighbour with SHD
IN1[0]
    ]
  ]

  |

  DN[ // destination node
    ::recv(s,msg).s::send().0
  |
    NH1[ // neighborhood of the destination node in direction 1
      0
    ]
  |
    NH2[ // neighborhood of the destination node in direction 2
      0
    ]
  |
    Hspeed[ // neighbours in high speed
      0
    ]
  |
    Lspeed[ // neighbours in low speed
      0
    ]

  | SHD[
0
    ]
  ]

  |

  IN1[ // intermediate node 1
    ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

```

```

/*
    if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)>d::send(IN1, msg).d::recv().0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(IN1, d, msg).n::recv().0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
for n::send(IN1, d, msg).n::recv().0

|
NH1[ // neighborhood of the intermediate node 1 in the direction 1
    IN2[0]
]
|
NH2[ // neighborhood of the intermediate node 1 in the direction 2
    0
]
|
Hspeed[ // neighbours in high speed
    IN2[0]
]
|
Lspeed[ // neighbours in low speed
    0
]

| SHD[
IN2[0]
]

```



```

]

|

IN2[ // intermediate node 2
    ::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
    if destination node d is neighbour to IN then send message to d.
*/
    <hasNb1(d)>d::send(IN2, msg).d::recv().0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD and highest speed.
*/

    | find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
    for n::send(IN2, d, msg).n::recv().0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 closer to d.
*/
    | find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
    for n::send(IN2, d, msg).n::recv().0

|

NH1[ // neighborhood of the intermediate node 2 in the direction 1
    0
]

|

NH2[ // neighborhood of the intermediate node 2 in the direction 2
    IN3[0]
]

|

Hspeed[ // neighbours in high speed
    IN3[0]
]

|

Lspeed[ // neighbours in low speed

```

```

0
]

| SHD[
IN3[0]
]
]

|

IN3[ // intermediate node 3
::recv(s,d,msg).s::send(). //receive a message and send acknowledgement to sender.

/*
if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)>d::send(IN3, msg).d::recv().0
/*
if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD(n) and highestSpeed(n)
for n::send(IN3, d, msg).n::recv().0
/*
if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD()) and hasNb1(n)
for n::send(IN3, d, msg).n::recv().0

|
NH1[ // neighborhood of the intermediate node 3 in the direction 1
DN[0]
]
|
NH2[ // neighborhood of the intermediate node 3 in the direction 2
0
]

```

```
|
  Hspeed[ // neighbours in high speed
    0
  ]
|
  Lspeed[ // neighbours in low speed
    0
  ]

| SHD[ // can be "one" or "zero".
0
  ]
]
```

Appendix C

Formal specification in the Calculus of Context-aware Ambients(CCA) for VSHDRP2

This appendix contains the full formal executable specification of the proposed routing protocol VSHDRP2. The specification contained three scenarios, namely *scenario1*, *scenario2* and *scenario3*.

Primary ambients:

- SN: source node
- DN: destination node
- IN1: intermediate node 1
- IN2: intermediate node 2

Directions:

- Direction 1: direction towards the destination node.

- Direction 2: opposite direction to direction 1.

Each node N contains the (secondary) ambients:

- NH1: contains the ids of all the neighbouring nodes, closer to the destination node, that are moving in direction 1.
- NH2: contains the ids of all the neighbouring nodes, closer to the destination node, that are moving in direction 2.
- Hspeedi: contains the ids of all the neighbouring nodes moving in high speed in direction 1 with SDH in zone i, i=1,2,3.
- Lspeedi: contains the ids of all the neighbouring nodes moving in low speed in direction 1 with SDH in zone i, i=1,2,3.
- SHDi: contains the ids of all the neighbouring in direction 1 with SHD in zone i, i=1,2,3.

Predicates:

- hasNb(n): n is a neighbour closer to destination (regardless its direction).
- hasNb1(n): n is a neighbour closer to destination in direction 1.
- hasNb2(n): n is a neighbour closer to destination in direction 2.
- noNb1(): no neighbours in direction 1.
- hasSHD1(n): n is a neighbour with SHD in zone 1.
- noSHD1(): no neighbours have SHD in zone 1.
- hasSHD2(n): n is a neighbour with SHD in zone 2.
- noSHD2(): no neighbours have SHD in zone 2.
- hasSHD3(n): n is a neighbour with SHD in zone 3.
- noSHD3(): no neighbours have SHD in zone 3.

- highSpeed1(n): n is a neighbour with high speed with SHD in zone 1.
- lowSpeed1(n): n is a neighbour with low speed with SHD in zone 1.
- highestSpeed1(n): n is a neighbour with the highest speed with SHD in zone 1.
- highSpeed2(n): n is a neighbour with high speed with SHD in zone 2.
- lowSpeed2(n): n is a neighbour with low speed with SHD in zone 2.
- highestSpeed2(n): n is a neighbour with the highest speed with SHD in zone 2.
- highSpeed3(n): n is a neighbour with high speed with SHD in zone 3.
- lowSpeed3(n): n is a neighbour with low speed with SHD in zone 3.
- highestSpeed3(n): n is a neighbour with the highest speed with SHD in zone 3.
- hasNbofNb1(n): n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction with SHD in zone 1.
- hasNbofNb2(n): n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1 with SHD in zone 2.
- hasNbofNb3(n): n is a neighbour in direction 2 that has a neighbour closer to destination and moving in direction 1 with SHD in zone 3.

/*

SCENARIO I

In this scenario the destination node DN is not neighbour to the source node SN.
Rather the neighborhood relation is defined as follows:

SN--->IN1

SN--->IN2

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
SN--->IN3
IN3--->DN

direction 1: SN, IN1, IN2, IN3
direction 2: IN4
Zone 1: IN1
Zone 2: IN2
Zone 3: IN3
*/

BEGIN_DECLS

def has(n) = this | n[true] | true
def at(n) = n[next (this | true)] | true
def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
def hasNb(n) = hasNb1(n) or hasNb2(n)

def highSpeed1(n) = somewhere(this | Hspeed1[true| n[true]] | true)
def lowSpeed1(n) = somewhere(this | Lspeed1[true| n[true]] | true)
def highestSpeed1(n) = highSpeed1(n) or (not (somewhere(this | Hspeed1[true | next true] | true))
and lowSpeed1(n))

def highSpeed2(n) = somewhere(this | Hspeed2[true| n[true]] | true)
def lowSpeed2(n) = somewhere(this | Lspeed2[true| n[true]] | true)
def highestSpeed2(n) = highSpeed2(n) or (not (somewhere(this | Hspeed2[true | next true] | true))
and lowSpeed2(n))

def highSpeed3(n) = somewhere(this | Hspeed3[true| n[true]] | true)
def lowSpeed3(n) = somewhere(this | Lspeed3[true| n[true]] | true)
def highestSpeed3(n) = highSpeed3(n) or (not (somewhere(this | Hspeed3[true | next true] | true))
and lowSpeed3(n))

def highSpeed4(n) = somewhere(this | Hspeed4[true| n[true]] | true)
def lowSpeed4(n) = somewhere(this | Lspeed4[true| n[true]] | true)
def highestSpeed4(n) = highSpeed4(n) or (not (somewhere(this | Hspeed4[true | next true] | true))
and lowSpeed4(n))

def hasSHD1(n) = somewhere (this | SHD1[ true | n[true]] | true)
def hasSHD2(n) = somewhere (this | SHD2[ true | n[true]] | true)
def hasSHD3(n) = somewhere (this | SHD3[ true | n[true]] | true)
```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
def hasSHD4(n) = somewhere (this | SHD4[ true | n[true]] | true)

def hasNb2(n) = hasNb2(n) and somewhere(n[ true | SHD1[true | next true]] | true)

def hasNb2(n) = hasNb2(n) and somewhere(n[ true | SHD2[true | next true]] | true)

def hasNb2(n) = hasNb2(n) and somewhere(n[ true | SHD3[true | next true]] | true)
def hasNb2(n) = hasNb2(n) and somewhere(n[ true | SHD4[true | next true]] | true)

def noNb1() = not somewhere(this | NH1[true| next true] | true)
def noSHD1() = not somewhere(this | SHD1[true| next true] | true)
def noSHD2() = not somewhere(this | SHD2[true| next true] | true)
def noSHD3() = not somewhere(this | SHD3[true| next true] | true)
def noSHD4() = not somewhere(this | SHD4[true| next true] | true)

//display code
//display congruence
END_DECLS
SN[ // source node

    new x new y new z new t

    x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]

    |
    /*
    if DN is neighbour to SN then send message to DN.
    */
    <hasNb(DN)> x#recv().y#recv().z#recv().DN::send(SN, packet).DN::recv(y).0
    |
    /*
    if DN is not neighbour to SN then look for an intermediate node in direction 1 with SHD1 and highest speed.
    */
    find n: not hasNb(DN) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
    for x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
    |
    /*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction SHD1 closer to DN.
    */
    find n: not hasNb(DN) and (noNb1() or noSHD1()) and hasNb2(n)
    for x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

    // zone 2
```


APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
    with SHD2 and highest speed.
*/
    find n: not hasNb(DN) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
    for y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction SHD2 closer to DN.
*/
    find n: not hasNb(DN) and (noNb1() or noSHD2()) and hasNb2(n)
    for y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 3
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/
    find n: not hasNb(DN) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
    for z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction SHD3 closer to DN.
*/
    find n: not hasNb(DN) and (noNb1() or noSHD3()) and hasNb3(n)
    for z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 4
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/
    find n: not hasNb(DN) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
    for t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
```

```

/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction SHD4 closer to DN.
*/

    find n: not hasNb(DN) and (noNb1() or noSHD4()) and hasNbofNb4(n)
    for t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

|

    NH1[ // neighborhood of the source node in direction 1 closer
// to the destination
        IN1[0] | IN2[0] | IN3[0]
    ]
|
    NH2[ // neighborhood of the source node in direction 2 closer
// to the destination
        0
    ]
|
    Hspeed1[ // neighbours in high speed in zone 1
        IN1[0]
    ]
|
    Lspeed1[ // neighbours in low speed in zone 1
        0
    ]
|
    Hspeed2[ // neighbours in high speed in zone 2
        IN2[0]
    ]
|
    Lspeed2[ // neighbours in low speed in zone 2
        0
    ]
|
    Hspeed3[ // neighbours in high speed in zone 3
        IN3[0]
    ]
|
    Lspeed3[ // neighbours in low speed in zone 3

```

```

        0
    ]
|
    Hspeed4[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed4[ // neighbours in low speed in zone 3
        0
    ]

| SHD1[ // neighbour with SHD in zone 1
    IN1[0]
]
| SHD2[ // neighbour with SHD in zone 2
IN2[0]
]
| SHD3[ // neighbour with SHD in zone 3
IN3[0]
]
| SHD4[ // neighbour with SHD in zone 3
0
]
]

|

DN[ // destination node
    !::recv(s,msg).s::send(ack).0
|
    NH1[ // neighborhood of the destination node in direction 1
        0
    ]
|
    NH2[ // neighborhood of the destination node in direction 2
        0
    ]
|
    Hspeed1[ // neighbours in high speed in zone 1
        0
    ]
]

```

```

|
  Lspeed1[ // neighbours in low speed in zone 1
    0
  ]
|
  Hspeed2[ // neighbours in high speed in zone 2
    0
  ]
|
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
|
  Hspeed3[ // neighbours in high speed in zone 3
    0
  ]
|
  Lspeed3[ // neighbours in low speed in zone 3
    0
  ]
|
  Hspeed4[ // neighbours in high speed in zone 3
    0
  ]
|
  Lspeed4[ // neighbours in low speed in zone 3
    0
  ]

| SHD1[ // neighbour with SHD in zone 1
0
  ]
| SHD2[ // neighbour with SHD in zone 2
0
  ]
| SHD3[ // neighbour with SHD in zone 3
0
  ]
| SHD4[ // neighbour with SHD in zone 3
0
  ]

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
]

|

IN1[ // intermediate node 1

  !::recv(s,d,msg).s::send(ack).new x new y new z new t//receive a message and send acknowledgement to sender.
    [@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
      |

/*
  if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN1, msg).d::recv(y).0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD1 and highest speed.
*/

  find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD1 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0

// zone 2
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD2 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD2 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1(n)
  for y#recv().n::send(IN1, d, msg).n::recv(y).0
```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
// Zone 3
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD4 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
  for  z#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD3 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNbofNb3(n)
  for  z#recv().n::send(IN1, d, msg).n::recv(y).0

// Zone 4
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD3 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
  for  t#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD4 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNbofNb4(n)
  for  t#recv().n::send(IN1, d, msg).n::recv(y).0

|
  NH1[ // neighborhood of the destination node in direction 1
    0
  ]
|
  NH2[ // neighborhood of the destination node in direction 2
    0
  ]
|
```

```

    Hspeed1[ // neighbours in high speed in zone 1
        0
    ]
|
    Lspeed1[ // neighbours in low speed in zone 1
        0
    ]
|
    Hspeed2[ // neighbours in high speed in zone 2
        0
    ]
|
    Lspeed2[ // neighbours in low speed in zone 2
        0
    ]
|
    Hspeed3[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed3[ // neighbours in low speed in zone 3
        0
    ]
|
    Hspeed4[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed4[ // neighbours in low speed in zone 3
        0
    ]
]

| SHD1[ // neighbour with SHD in zone 1
0
]
| SHD2[ // neighbour with SHD in zone 2
0
]
| SHD3[ // neighbour with SHD in zone 3
0
]

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
| SHD4[ // neighbour with SHD in zone 3
0
]
]

|

IN2[ // intermediate node 2
!::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*
    if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN2, msg).d::recv(y).0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD1 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1(n)
for x#recv().n::send(IN2, d, msg).n::recv(y).0

// zone 2
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD2 closer to d.
*/
```


APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
for y#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    0
]
|
NH2[ // neighborhood of the destination node in direction 2
```

```

        0
    ]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2
    0
]
|
Lspeed2[ // neighbours in low speed in zone 2
    0
]
|
Hspeed3[ // neighbours in high speed in zone 3
    0
]
|
Lspeed3[ // neighbours in low speed in zone 3
    0
]

| SHD1[ // neighbour with SHD in zone 1
0
]
| SHD2[ // neighbour with SHD in zone 2
0
]
| SHD3[ // neighbour with SHD in zone 3
0
]

|

IN3[ // intermediate node 3

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
!::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*
  if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN3, msg).d::recv(y).0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN3, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD1 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
for x#recv().n::send(IN3, d, msg).n::recv(y).0

// zone 2
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN3, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD2 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
for y#recv().n::send(IN3, d, msg).n::recv(y).0

// Zone 3
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD4 and highest speed.
*/
```

```

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN3, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
for z#recv().n::send(IN3, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN3, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for t#recv().n::send(IN3, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    DN[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1

```

```

        0
    ]
|
    Hspeed2[ // neighbours in high speed in zone 2
        0
    ]
|
    Lspeed2[ // neighbours in low speed in zone 2
        0
    ]
|
    Hspeed3[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed3[ // neighbours in low speed in zone 3
        0
    ]

| SHD1[ // neighbour with SHD in zone 1
0
    ]
| SHD2[ // neighbour with SHD in zone 2
0
    ]
| SHD3[ // neighbour with SHD in zone 3
0
    ]

|

IN4[ // intermediate node 4
    !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
    x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*
    if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN4, msg).d::recv(y).0
/*

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN4, d, msg).n::recv(y).0
/*

    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD1 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
for x#recv().n::send(IN4, d, msg).n::recv(y).0

// zone 2
/*

    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN4, d, msg).n::recv(y).0
/*

    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD2 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
for y#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 3
/*

    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN4, d, msg).n::recv(y).0
/*

    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
```

```

for  z#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for  t#recv().n::send(IN4, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for  t#recv().n::send(IN4, d, msg).n::recv(y).0


|
NH1[ // neighborhood of the destination node in direction 1
    IN3[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2
    0
]
|
Lspeed2[ // neighbours in low speed in zone 2

```

```

        0
    ]
    |
    Hspeed3[ // neighbours in high speed in zone 3
        IN3[0]
    ]
    |
    Lspeed3[ // neighbours in low speed in zone 3
        0
    ]
    |
    Hspeed4[ // neighbours in high speed in zone 3
        0
    ]
    |
    Lspeed4[ // neighbours in low speed in zone 3
        0
    ]

    | SHD1[ // neighbour with SHD in zone 1
0
    ]
    | SHD2[ // neighbour with SHD in zone 2
0
    ]
    | SHD3[ // neighbour with SHD in zone 3
IN3[0]
    ]
    | SHD4[ // neighbour with SHD in zone 3
0
    ]
]

/*

```

SCENARIO II

In this scenario the destination node DN is not neighbour to the source node SN.
Rather the neighborhood relation is defined as follows:


```

SN--->IN1
SN--->IN2
SN--->IN4
IN4--->IN3
IN3--->DN

direction 1: SN, IN1, IN2, IN3
direction 2: IN4
Zone 1: IN1
Zone 2: IN2
Zone 3: IN3
*/

BEGIN_DECLS

def has(n) = this | n[true] | true
def at(n) = n[next (this | true)] | true
def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
def hasNb(n) = hasNb1(n) or hasNb2(n)

def highSpeed1(n) = somewhere(this | Hspeed1[true| n[true]] | true)
def lowSpeed1(n) = somewhere(this | Lspeed1[true| n[true]] | true)
def highestSpeed1(n) = highSpeed1(n) or (not (somewhere(this | Hspeed1[true | next true] | true))
and lowSpeed1(n))

def highSpeed2(n) = somewhere(this | Hspeed2[true| n[true]] | true)
def lowSpeed2(n) = somewhere(this | Lspeed2[true| n[true]] | true)
def highestSpeed2(n) = highSpeed2(n) or (not (somewhere(this | Hspeed2[true | next true] | true))
and lowSpeed2(n))

def highSpeed3(n) = somewhere(this | Hspeed3[true| n[true]] | true)
def lowSpeed3(n) = somewhere(this | Lspeed3[true| n[true]] | true)
def highestSpeed3(n) = highSpeed3(n) or (not (somewhere(this | Hspeed3[true | next true] | true))
and lowSpeed3(n))

def highSpeed4(n) = somewhere(this | Hspeed4[true| n[true]] | true)
def lowSpeed4(n) = somewhere(this | Lspeed4[true| n[true]] | true)
def highestSpeed4(n) = highSpeed4(n) or (not (somewhere(this | Hspeed4[true | next true] | true))
and lowSpeed4(n))

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```

def hasSHD1(n) = somewhere (this | SHD1[ true | n[true]] | true)
def hasSHD2(n) = somewhere (this | SHD2[ true | n[true]] | true)
def hasSHD3(n) = somewhere (this | SHD3[ true | n[true]] | true)
def hasSHD4(n) = somewhere (this | SHD4[ true | n[true]] | true)
def hasNbofNb1(n) = hasNb2(n) and somewhere(n[ true | SHD1[true | next true]] | true)

def hasNbofNb2(n) = hasNb2(n) and somewhere(n[ true | SHD2[true | next true]] | true)

def hasNbofNb3(n) = hasNb2(n) and somewhere(n[ true | SHD3[true | next true]] | true)
def hasNbofNb4(n) = hasNb2(n) and somewhere(n[ true | SHD4[true | next true]] | true)

def noNb1() = not somewhere(this | NH1[true| next true] | true)
def noSHD1() = not somewhere(this | SHD1[true| next true] | true)
def noSHD2() = not somewhere(this | SHD2[true| next true] | true)
def noSHD3() = not somewhere(this | SHD3[true| next true] | true)
def noSHD4() = not somewhere(this | SHD4[true| next true] | true)

//display code
//display congruence
END_DECLS
SN[ // source node

new x new y new z new t

x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*
  if DN is neighbour to SN then send message to DN.
*/
<hasNb(DN)> x#recv().y#recv().z#recv().DN::send(SN, packet).DN::recv(y).0
|
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD1 and highest speed.
*/
find n: not hasNb(DN) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD1 closer to DN.

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
*/
  find n: not hasNb(DN) and (noNb1() or noSHD1()) and hasNbofNb1(n)
for  x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 2
|
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD2 and highest speed.
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for  y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD2 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD2()) and hasNbofNb2(n)
for  y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 3
|
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD3 and highest speed.
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for  z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD3 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD3()) and hasNbofNb3(n)
for  z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 4
|
/*
  if DN is not neighbour to SN then look for an intermediate node in direction 1
```

```

with SHD4 and highest speed.
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for  t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD4 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD4()) and hasNb4(n)
for  t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

|

  NH1[ // neighborhood of the source node in direction 1 closer
// to the destination
    IN1[0] | IN2[0]
  ]
|
  NH2[ // neighborhood of the source node in direction 2 closer
// to the destination
    IN4[0]
  ]
|
  Hspeed1[ // neighbours in high speed in zone 1
    IN1[0]
  ]
|
  Lspeed1[ // neighbours in low speed in zone 1
    0
  ]
|
  Hspeed2[ // neighbours in high speed in zone 2
    IN2[0]
  ]
|
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
|

```

```

Hspeed3[ // neighbours in high speed in zone 3
  0
]
|
Lspeed3[ // neighbours in low speed in zone 3
  0
]
|
Hspeed4[ // neighbours in high speed in zone 3
  0
]
|
Lspeed4[ // neighbours in low speed in zone 3
  0
]

| SHD1[ // neighbour with SHD in zone 1
  IN1[0]
]
| SHD2[ // neighbour with SHD in zone 2
  IN2[0]
]
| SHD3[ // neighbour with SHD in zone 3
  0
]
| SHD4[ // neighbour with SHD in zone 3
  0
]

|

DN[ // destination node
  !::recv(s,msg).s::send(ack).0
]
|
NH1[ // neighborhood of the destination node in direction 1
  0
]
|
NH2[ // neighborhood of the destination node in direction 2
  0
]

```

```

    ]
  |
    Hspeed1[ // neighbours in high speed in zone 1
      0
    ]
  |
    Lspeed1[ // neighbours in low speed in zone 1
      0
    ]
  |
    Hspeed2[ // neighbours in high speed in zone 2
      0
    ]
  |
    Lspeed2[ // neighbours in low speed in zone 2
      0
    ]
  |
    Hspeed3[ // neighbours in high speed in zone 3
      0
    ]
  |
    Lspeed3[ // neighbours in low speed in zone 3
      0
    ]
  |
    Hspeed4[ // neighbours in high speed in zone 3
      0
    ]
  |
    Lspeed4[ // neighbours in low speed in zone 3
      0
    ]
  ]

  | SHD1[ // neighbour with SHD in zone 1
0
  ]
  | SHD2[ // neighbour with SHD in zone 2
0
  ]
  | SHD3[ // neighbour with SHD in zone 3

```

```

0
  ]
  | SHD4[ // neighbour with SHD in zone 3
0
  ]
]

|

IN1[ // intermediate node 1
  !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
  x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
  |
/*
  if destination node d is neighbour to IN then send message to d.
*/
  <hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN1, msg).d::recv(y).0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD1 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD1 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0

// zone 2
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD2 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
which has a neighbour in direction 1 with SHD2 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb2(n)
for y#recv().n::send(IN1, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN1, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb3(n)
for z#recv().n::send(IN1, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN1, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb4(n)
for t#recv().n::send(IN1, d, msg).n::recv(y).0

|
  NH1[ // neighborhood of the destination node in direction 1
    0
  ]
|
```



```

    NH2[ // neighborhood of the destination node in direction 2
      0
    ]
  |
  Hspeed1[ // neighbours in high speed in zone 1
    0
  ]
  |
  Lspeed1[ // neighbours in low speed in zone 1
    0
  ]
  |
  Hspeed2[ // neighbours in high speed in zone 2
    0
  ]
  |
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
  |
  Hspeed3[ // neighbours in high speed in zone 3
    0
  ]
  |
  Lspeed3[ // neighbours in low speed in zone 3
    0
  ]
  |
  Hspeed4[ // neighbours in high speed in zone 3
    0
  ]
  |
  Lspeed4[ // neighbours in low speed in zone 3
    0
  ]

  | SHD1[ // neighbour with SHD in zone 1
0
  ]
  | SHD2[ // neighbour with SHD in zone 2
0

```

```

    ]
    | SHD3[ // neighbour with SHD in zone 3
0
    ]
    | SHD4[ // neighbour with SHD in zone 3
0
    ]
]

|

IN2[ // intermediate node 2
  !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
  x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
  |
/*
    if destination node d is neighbour to IN then send message to d.
*/
  <hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN2, msg).d::recv(y).0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD1 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD1 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1(n)
  for x#recv().n::send(IN2, d, msg).n::recv(y).0

// zone 2
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD2 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN2, d, msg).n::recv(y).0

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD2 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
for y#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
```

```

        0
    ]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2
    0
]
|
Lspeed2[ // neighbours in low speed in zone 2
    0
]
|
Hspeed3[ // neighbours in high speed in zone 3
    0
]
|
Lspeed3[ // neighbours in low speed in zone 3
    0
]

| SHD1[ // neighbour with SHD in zone 1
0
]
| SHD2[ // neighbour with SHD in zone 2
0
]
| SHD3[ // neighbour with SHD in zone 3
0
]
]

```

```

|

IN3[ // intermediate node 3

  !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
  x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]

  |

  /*
    if destination node d is neighbour to IN then send message to d.
  */

  <hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN3, msg).d::recv(y).0

  /*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD1 and highest speed.
  */

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN3, d, msg).n::recv(y).0

  /*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD1 closer to d.
  */

  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
  for x#recv().n::send(IN3, d, msg).n::recv(y).0

  // zone 2

  /*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD2 and highest speed.
  */

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN3, d, msg).n::recv(y).0

  /*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD2 closer to d.
  */

  | find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
  for y#recv().n::send(IN3, d, msg).n::recv(y).0

  // Zone 3

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for  z#recv().n::send(IN3, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
for  z#recv().n::send(IN3, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for  t#recv().n::send(IN3, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for  t#recv().n::send(IN3, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    DN[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
```

```

        0
    ]
|
    Lspeed1[ // neighbours in low speed in zone 1
        0
    ]
|
    Hspeed2[ // neighbours in high speed in zone 2
        0
    ]
|
    Lspeed2[ // neighbours in low speed in zone 2
        0
    ]
|
    Hspeed3[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed3[ // neighbours in low speed in zone 3
        0
    ]
]

| SHD1[ // neighbour with SHD in zone 1
0
    ]
| SHD2[ // neighbour with SHD in zone 2
0
    ]
| SHD3[ // neighbour with SHD in zone 3
0
    ]
]

|

IN4[ // intermediate node 4
    !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
    x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
    if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN4, msg).d::recv(y).0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN4, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD1 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNbofNb1(n)
for x#recv().n::send(IN4, d, msg).n::recv(y).0

// zone 2
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN4, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD2 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNbofNb2(n)
for y#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN4, d, msg).n::recv(y).0
/*
```



```

    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/
    | find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb3(n)
    for z#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

    | find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
    for t#recv().n::send(IN4, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

    | find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb4(n)
    for t#recv().n::send(IN4, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    IN3[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2

```

```

    0
  ]
|
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
|
  Hspeed3[ // neighbours in high speed in zone 3
    IN3[0]
  ]
|
  Lspeed3[ // neighbours in low speed in zone 3
    0
  ]
|
  Hspeed4[ // neighbours in high speed in zone 3
    0
  ]
|
  Lspeed4[ // neighbours in low speed in zone 3
    0
  ]

| SHD1[ // neighbour with SHD in zone 1
0
  ]
| SHD2[ // neighbour with SHD in zone 2
0
  ]
| SHD3[ // neighbour with SHD in zone 3
IN3[0]
  ]
| SHD4[ // neighbour with SHD in zone 3
0
  ]
]

/*

```

SCENARIO III

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

In this scenario the destination node DN is a neighbour to the source node SN.

SN--->DN

direction 1: SN, IN1, IN2, IN3

direction 2: IN4

Zone 1: IN1

Zone 2: IN2

Zone 3: IN3

*/

BEGIN_DECLS

```
def has(n) = this | n[true] | true
```

```
def at(n) = n[next (this | true)] | true
```

```
def hasNb1(n) = somewhere(this | NH1[true| n[true]] | true)
```

```
def hasNb2(n) = somewhere(this | NH2[true| n[true]] | true)
```

```
def hasNb(n) = hasNb1(n) or hasNb2(n)
```

```
def highSpeed1(n) = somewhere(this | Hspeed1[true| n[true]] | true)
```

```
def lowSpeed1(n) = somewhere(this | Lspeed1[true| n[true]] | true)
```

```
def highestSpeed1(n) = highSpeed1(n) or (not (somewhere(this | Hspeed1[true | next true] | true))  
and lowSpeed1(n))
```

```
def highSpeed2(n) = somewhere(this | Hspeed2[true| n[true]] | true)
```

```
def lowSpeed2(n) = somewhere(this | Lspeed2[true| n[true]] | true)
```

```
def highestSpeed2(n) = highSpeed2(n) or (not (somewhere(this | Hspeed2[true | next true] | true))  
and lowSpeed2(n))
```

```
def highSpeed3(n) = somewhere(this | Hspeed3[true| n[true]] | true)
```

```
def lowSpeed3(n) = somewhere(this | Lspeed3[true| n[true]] | true)
```

```
def highestSpeed3(n) = highSpeed3(n) or (not (somewhere(this | Hspeed3[true | next true] | true))  
and lowSpeed3(n))
```

```
def highSpeed4(n) = somewhere(this | Hspeed4[true| n[true]] | true)
```

```
def lowSpeed4(n) = somewhere(this | Lspeed4[true| n[true]] | true)
```

```
def highestSpeed4(n) = highSpeed4(n) or (not (somewhere(this | Hspeed4[true | next true] | true))  
and lowSpeed4(n))
```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```

def hasSHD1(n) = somewhere (this | SHD1[ true | n[true]] | true)
def hasSHD2(n) = somewhere (this | SHD2[ true | n[true]] | true)
def hasSHD3(n) = somewhere (this | SHD3[ true | n[true]] | true)
def hasSHD4(n) = somewhere (this | SHD4[ true | n[true]] | true)
def hasNb1(n) = hasNb2(n) and somewhere(n[ true | SHD1[true | next true]] | true)

def hasNb2(n) = hasNb2(n) and somewhere(n[ true | SHD2[true | next true]] | true)

def hasNb3(n) = hasNb2(n) and somewhere(n[ true | SHD3[true | next true]] | true)
def hasNb4(n) = hasNb2(n) and somewhere(n[ true | SHD4[true | next true]] | true)

def noNb1() = not somewhere(this | NH1[true| next true] | true)
def noSHD1() = not somewhere(this | SHD1[true| next true] | true)
def noSHD2() = not somewhere(this | SHD2[true| next true] | true)
def noSHD3() = not somewhere(this | SHD3[true| next true] | true)
def noSHD4() = not somewhere(this | SHD4[true| next true] | true)

//display code
//display congruence
END_DECLS
SN[ // source node

    new x new y new z new t

    x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]

    |
    /*
    if DN is neighbour to SN then send message to DN.
    */
    <hasNb(DN)> x#recv().y#recv().z#recv().DN::send(SN, packet).DN::recv(y).0
    |
    /*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
    with SHD1 and highest speed.
    */
    find n: not hasNb(DN) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
    for x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
    |
    /*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction SHD1 closer to DN.
    */

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
    find n: not hasNb(DN) and (noNb1() or noSHD1()) and hasNbofNb1(n)
for   x#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 2
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD2 and highest speed.
*/
    find n: not hasNb(DN) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for   y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD2 closer to DN.
*/
    find n: not hasNb(DN) and (noNb1() or noSHD2()) and hasNbofNb2(n)
for   y#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 3
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD3 and highest speed.
*/
    find n: not hasNb(DN) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for   z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD3 closer to DN.
*/
    find n: not hasNb(DN) and (noNb1() or noSHD3()) and hasNbofNb3(n)
for   z#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN

// zone 4
|
/*
    if DN is not neighbour to SN then look for an intermediate node in direction 1
with SHD4 and highest speed.
```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
*/
  find n: not hasNb(DN) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for  t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction SHD4 closer to DN.
*/
  find n: not hasNb(DN) and (noNb1() or noSHD4()) and hasNbofNb4(n)
for  t#recv().n::send(SN, DN, packet).n::recv(y).0 // DN is not neighbour to SN
|

  NH1[ // neighborhood of the source node in direction 1 closer
// to the destination
    DN[0] | IN1[0] | IN2[0] | IN5[0]
  ]
|
  NH2[ // neighborhood of the source node in direction 2 closer
// to the destination
    IN4[0]
  ]
|
  Hspeed1[ // neighbours in high speed in zone 1
    IN1[0]
  ]
|
  Lspeed1[ // neighbours in low speed in zone 1
    IN5[0]
  ]
|
  Hspeed2[ // neighbours in high speed in zone 2
    IN2[0]
  ]
|
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
|
  Hspeed3[ // neighbours in high speed in zone 3
```

```

        0
    ]
|
    Lspeed3[ // neighbours in low speed in zone 3
        0
    ]
|
    Hspeed4[ // neighbours in high speed in zone 3
        0
    ]
|
    Lspeed4[ // neighbours in low speed in zone 3
        0
    ]

| SHD1[ // neighbour with SHD in zone 1
    IN1[0] | IN5[0]
]
| SHD2[ // neighbour with SHD in zone 2
IN2[0]
]
| SHD3[ // neighbour with SHD in zone 3
0
]
| SHD4[ // neighbour with SHD in zone 3
0
]
]

|

DN[ // destination node
    !::recv(s,msg).s::send(ack).0
|
    NH1[ // neighborhood of the destination node in direction 1
        0
    ]
|
    NH2[ // neighborhood of the destination node in direction 2
        0
    ]
]

```

```

|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2
    0
]
|
Lspeed2[ // neighbours in low speed in zone 2
    0
]
|
Hspeed3[ // neighbours in high speed in zone 3
    0
]
|
Lspeed3[ // neighbours in low speed in zone 3
    0
]
|
Hspeed4[ // neighbours in high speed in zone 3
    0
]
|
Lspeed4[ // neighbours in low speed in zone 3
    0
]

| SHD1[ // neighbour with SHD in zone 1
0
]
| SHD2[ // neighbour with SHD in zone 2
0
]
| SHD3[ // neighbour with SHD in zone 3
0

```



```

    ]
  | SHD4[ // neighbour with SHD in zone 3
0
    ]
]

|

IN1[ // intermediate node 1

  !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
  x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
  |
/*
  if destination node d is neighbour to IN then send message to d.
*/
  <hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN1, msg).d::recv(y).0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD1 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD1 closer to d.
*/
  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
  for x#recv().n::send(IN1, d, msg).n::recv(y).0

// zone 2
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD2 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN1, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD2 closer to d.

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
*/
| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNbofNb2(n)
for y#recv().n::send(IN1, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN1, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNbofNb3(n)
for z#recv().n::send(IN1, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN1, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNbofNb4(n)
for t#recv().n::send(IN1, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    0
]
|
NH2[ // neighborhood of the destination node in direction 2
```

```

        0
      ]
    |
    Hspeed1[ // neighbours in high speed in zone 1
      0
    ]
    |
    Lspeed1[ // neighbours in low speed in zone 1
      0
    ]
    |
    Hspeed2[ // neighbours in high speed in zone 2
      0
    ]
    |
    Lspeed2[ // neighbours in low speed in zone 2
      0
    ]
    |
    Hspeed3[ // neighbours in high speed in zone 3
      0
    ]
    |
    Lspeed3[ // neighbours in low speed in zone 3
      0
    ]
    |
    Hspeed4[ // neighbours in high speed in zone 3
      0
    ]
    |
    Lspeed4[ // neighbours in low speed in zone 3
      0
    ]
  ]

| SHD1[ // neighbour with SHD in zone 1
0
]
| SHD2[ // neighbour with SHD in zone 2
0
]

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
| SHD3[ // neighbour with SHD in zone 3
0
|
| SHD4[ // neighbour with SHD in zone 3
0
|
]

|

IN2[ // intermediate node 2

!::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
|
/*
    if destination node d is neighbour to IN then send message to d.
*/
<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN2, msg).d::recv(y).0
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD1 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
for x#recv().n::send(IN2, d, msg).n::recv(y).0

// zone 2
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN2, d, msg).n::recv(y).0
/*
```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD2 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb2(n)
for y#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 3
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb3(n)
for z#recv().n::send(IN2, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb4(n)
for t#recv().n::send(IN2, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
0
```

```

    ]
  |
    NH2[ // neighborhood of the destination node in direction 2
      0
    ]
  |
    Hspeed1[ // neighbours in high speed in zone 1
      0
    ]
  |
    Lspeed1[ // neighbours in low speed in zone 1
      0
    ]
  |
    Hspeed2[ // neighbours in high speed in zone 2
      0
    ]
  |
    Lspeed2[ // neighbours in low speed in zone 2
      0
    ]
  |
    Hspeed3[ // neighbours in high speed in zone 3
      0
    ]
  |
    Lspeed3[ // neighbours in low speed in zone 3
      0
    ]
  ]

  | SHD1[ // neighbour with SHD in zone 1
0
  ]
  | SHD2[ // neighbour with SHD in zone 2
0
  ]
  | SHD3[ // neighbour with SHD in zone 3
0
  ]
]

```

```

|

IN3[ // intermediate node 3

!::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]

|

/*
    if destination node d is neighbour to IN then send message to d.
*/

<hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN3, msg).d::recv(y).0

/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD1 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
for x#recv().n::send(IN3, d, msg).n::recv(y).0

/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD1 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
for x#recv().n::send(IN3, d, msg).n::recv(y).0

// zone 2

/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD2 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
for y#recv().n::send(IN3, d, msg).n::recv(y).0

/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD2 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
for y#recv().n::send(IN3, d, msg).n::recv(y).0

// Zone 3

/*

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD4 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
for  z#recv().n::send(IN3, d, msg).n::recv(y).0
/*

    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD3 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb1ofNb3(n)
for  z#recv().n::send(IN3, d, msg).n::recv(y).0

// Zone 4
/*

    if d is not neighbour to IN then look for an intermediate node in direction 1
with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for  t#recv().n::send(IN3, d, msg).n::recv(y).0
/*

    if no such intermediate nodes, look for an intermediate node in direction 2
which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb1ofNb4(n)
for  t#recv().n::send(IN3, d, msg).n::recv(y).0

|
NH1[ // neighborhood of the destination node in direction 1
    DN[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
```



```

    ]
  |
    Lspeed1[ // neighbours in low speed in zone 1
      0
    ]
  |
    Hspeed2[ // neighbours in high speed in zone 2
      0
    ]
  |
    Lspeed2[ // neighbours in low speed in zone 2
      0
    ]
  |
    Hspeed3[ // neighbours in high speed in zone 3
      0
    ]
  |
    Lspeed3[ // neighbours in low speed in zone 3
      0
    ]
  ]

  | SHD1[ // neighbour with SHD in zone 1
0
  ]
  | SHD2[ // neighbour with SHD in zone 2
0
  ]
  | SHD3[ // neighbour with SHD in zone 3
0
  ]
]

|

IN4[ // intermediate node 4
  !::recv(s,d,msg).s::send(ack).new x new y new z new t //receive a message and send acknowledgement to sender.
  x[@send().0] | y[@send().0] | z[@send().0] | t[@send().0]
  |
/*
  if destination node d is neighbour to IN then send message to d.

```

```

*/
  <hasNb1(d)> x#recv().y#recv().z#recv().d::send(IN4, msg).d::recv(y).0
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD1 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD1(n) and highestSpeed1(n)
  for x#recv().n::send(IN4, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD1 closer to d.
*/

  | find n: not hasNb(d) and (noNb1() or noSHD1()) and hasNb1ofNb1(n)
  for x#recv().n::send(IN4, d, msg).n::recv(y).0

// zone 2
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD2 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD2(n) and highestSpeed2(n)
  for y#recv().n::send(IN4, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2
  which has a neighbour in direction 1 with SHD2 closer to d.
*/

  | find n: not hasNb(d) and (noNb1() or noSHD2()) and hasNb1ofNb2(n)
  for y#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 3
/*
  if d is not neighbour to IN then look for an intermediate node in direction 1
  with SHD4 and highest speed.
*/

  | find n: not hasNb1(d) and hasNb1(n) and hasSHD3(n) and highestSpeed3(n)
  for z#recv().n::send(IN4, d, msg).n::recv(y).0
/*
  if no such intermediate nodes, look for an intermediate node in direction 2

```

APPENDIX C. FORMAL SPECIFICATION IN THE CALCULUS OF CONTEXT-AWARE AMBIENTS(CCA) FOR VSHDRP2

```
which has a neighbour in direction 1 with SHD3 closer to d.
*/
| find n: not hasNb(d) and (noNb1() or noSHD3()) and hasNb3(n)
for z#recv().n::send(IN4, d, msg).n::recv(y).0

// Zone 4
/*
    if d is not neighbour to IN then look for an intermediate node in direction 1
    with SHD3 and highest speed.
*/

| find n: not hasNb1(d) and hasNb1(n) and hasSHD4(n) and highestSpeed4(n)
for t#recv().n::send(IN4, d, msg).n::recv(y).0
/*
    if no such intermediate nodes, look for an intermediate node in direction 2
    which has a neighbour in direction 1 with SHD4 closer to d.
*/

| find n: not hasNb(d) and (noNb1() or noSHD4()) and hasNb4(n)
for t#recv().n::send(IN4, d, msg).n::recv(y).0


|
NH1[ // neighborhood of the destination node in direction 1
    IN3[0]
]
|
NH2[ // neighborhood of the destination node in direction 2
    0
]
|
Hspeed1[ // neighbours in high speed in zone 1
    0
]
|
Lspeed1[ // neighbours in low speed in zone 1
    0
]
|
Hspeed2[ // neighbours in high speed in zone 2
```

```

    0
  ]
|
  Lspeed2[ // neighbours in low speed in zone 2
    0
  ]
|
  Hspeed3[ // neighbours in high speed in zone 3
    IN3[0]
  ]
|
  Lspeed3[ // neighbours in low speed in zone 3
    0
  ]
|
  Hspeed4[ // neighbours in high speed in zone 3
    0
  ]
|
  Lspeed4[ // neighbours in low speed in zone 3
    0
  ]

| SHD1[ // neighbour with SHD in zone 1
0
  ]
| SHD2[ // neighbour with SHD in zone 2
0
  ]
| SHD3[ // neighbour with SHD in zone 3
IN3[0]
  ]
| SHD4[ // neighbour with SHD in zone 3
0
  ]
]

```